A Qualitative Discretization for Two-Body Contacts

Amitabha Mukerjee and **Manish Agarwal**
Center for Robotics
Indian Institute of Technology
Kanpur, India.
e-mail: amit@iitk.ernet.in

Praveen Bhatia
Robotics Group
Centre for Artificial Intelligence and Robotics
Bangalore, India,
e-mail: pbhatia@cair.emet.in

Abstract

In this work we use contact alignments as qualitative landmarks to discretize the relative motion between two 3D objects. We use assembly planning as a sample domain, and address the question of obtaining the assembly blocking graphs from the geometry and the motion constraints. Starting from a geometrical description of the objects we characterize contacts involving topologically distinct feature sets, called *contact formations (CF)*, and obtain a qualitative decomposition of the configuration space based on CFs. We show how standard algorithms for finding the configuration-space routinely discard CF information, and how these can be extracted at no additional computational cost. Finally we show how CFs can be used to generate assembly solutions and for correcting jamming and other assembly er-

Keywords: Qualitative Spatial Reasoning, Assembly Planning, Contact Formations

1 Introduction

Contact preserving motions between spatial objects is a question that is central to qualitative spatial reasoning [I], constructing assembly plans [7], kinematics [10], fine motion planning [8], gross motion planning [12], and other tasks. In this work we use contact alignments as qualitative landmarks to discretize the relative motion between 3D objects. We focus on assembly planning as an example domain, and show how to obtain a search space from geometric and motion constraint considerations and provide algorithms for the 2-body motion problem in any dimension.

Consider a sequence of motions such as: "Incline the peg and trail it along the top face until you find the hole. Align the edges, straighten and insert it. If jammed, you may be touching a side wall." Intuitively this involves a decomposition of the motion space between the peg and the hole into regions such as "inclined-peg-top-face", "aligned-edges", "straight-peg", "touching-side-wall" etc. Much of AI is concerned with the problem of how to develop this type of discretization - in the assembly problem, for example, the question is: given two objects in contact, how can one characterize the relative motion between them into regions that are *qualitatively* distinct?

Such a discretization is crucial to reducing the size of the search space for fine motions, which in a general configuration space approach results in a topological partition with an exponential number of manifolds [5], and in practice, it is unlikely (PSPACE hard [18]) that one can test the feasibility of the large number of proposals made by the typical assembly sequencers [3]. Computationally such a discretization reduces the search from the continuous domain of part configurations to a finite set of contact regions. Furthermore, associated characteristics of each region help in correcting errors - e.g. when jammed, the reason can be found by investigating neighbouring contacts (see [8] for an error recovery application).

Yet, surprisingly little work has been done on the process of finding such discretizations for general geometries. In constructing assembly plans, one assumes the existence of such a decomposition: AND/OR graphs [7], Precedence graphs [4], Backward Assembly Planning [13], Assembly Constraint Graph [20], etc. The closest approach to handling general geometries is that of Wilson and Latombe [19], which extends the blocking graph model to identify a discretization in the translation space in two dimensions. The model works only for assembly, and is not useful in spatial reasoning, kinematics, or other tasks. Also, it is restricted to 2D and does not handle rotations. Another class of models used in qualitative kinematics, also approaches this work by forming discretizations along the boundaries of the configuration [9]. However, these approaches discard the internal points of the C-space, which are necessary for recording relative motions between faces in 3-D for example, and do not use contact analysis as a mechanism for capturing the discretization in the spatial positions as such. In terms of the assembly problem, some specialized geometries belonging to classes like nuts on bolts can be analyzed using knowledge-based techniques, but this cannot extend to the general class of geometries, nor can the correctness of the knowledge database be verified without more general purpose algorithms.

In what sense is this a "qualitative" discretization of

the spatial 2-body problem? Many applications of QR involve parameter spaces that reduce to one-dimensional points (-,0,+) or intervals on the real line ([2]). We were encouraged in this work by qualitative models that discretize visibility regions of an object into "aspect graphs" [11], and by recent work on modeling 3D positions using a qualitative-quantitative hybrid [15]. Earlier work in modeling contacts, using the name *Contact Formations* [8], uses boundary features such as edges, vertices, faces, to generate a search space in error recovery situations. Other models of qualitative spatial reasoning have attempted connections among spatial objects using rectangular enclosures or other approximations [6], and have failed to define a clearly defined, usable set of parameters for qualitatively discriminating the interactions. Cognitive studies of assembly tasks also seem to indicate the importance of "haptic landmarks" which are formed while playing around with the parts prior to actual assembly. Thus just as the alignment orientation provide "visual landmarks" in constructing a qualitative model for recognition, the contact formations provide a qualitative set of haptic landmarks for fine motion, kinematics, and other tasks. In connection with recognizing landmarks, the new statistical learning techniques for finding landmarks is of particular note [17].

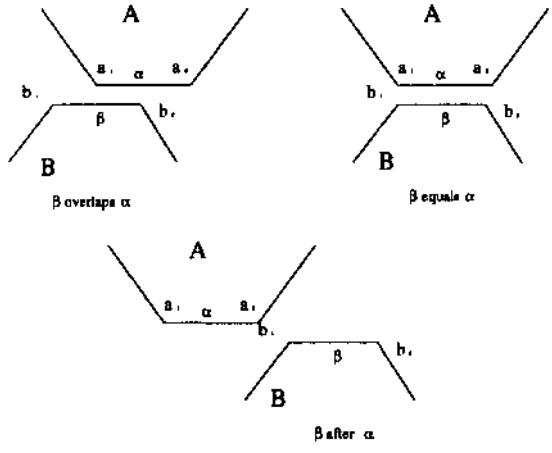

Figure 1: *Translational Contact Motion.* As B slides from left to right, the relative position of B w.r.t. A is characterized by the interval relations between a_1a_2 and b_1b_2: *overlaps* to *equal* to *after*.

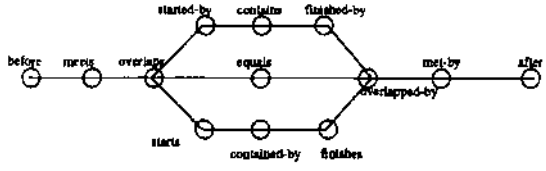

Figure 2: Sliding contact in one dimension: Transition graph between two intervals [Allen]

We claim that haptic landmarks are obtained by alignments between boundary features of the objects (such as that involving an edge from one 3D object and a face in another). Such alignments mark regions of change in the contact forces as the set of surface features in contact are altered. For example, considering purely sliding motion, we observe that the transitions closely resemble the interval relations well-known in temporal logic and one-dimensional qualitative spatial reasoning (fig. 1). As the edge B slides along the edge *a* and its interval relations change, so does the set of contacts occurring between boundary features in A and B. Thus the B *overlaps a* configuration is characterized by the following contacts:

a) vertex a_1 and edge β
b) edge α and vertex b_2
c) edge α and edge β

This is represented as a contact formation with three elemental contacts: $\{(a_1,\beta),(\alpha,b_2),(\alpha,\beta)\}$ The entire set of transitions possible in such motion is illustrated in the transition graph (Fig 2) [14], where the tri-furcation indicates the three possibilities with relative size:

(a) B smaller than A (starts) $(|\alpha| > |\beta|)$ $(\alpha,\beta),(a_1,b_1),(\alpha,b_2)$

(b) B equal than B (equals) $(|\alpha| = |\beta|)$ $(a_1,b_1),(\alpha,\beta),(a_2,b_2)$

(c) B longer than A (started-by) (|al| < \B\) : $(\alpha,\beta),(a_1,b_1),(a_2,\beta)$

Pure sliding motions can be modeled in this way, but what about rotations? One way to map the linear interval problem to a cyclic space (fig. 3), where alignments between surface features such as vertices and edges provide decompositions of qualitative interest. Here also, intervals exhibit the same relations except that "after" and "before" in the transition graph of fig. 2 have been joined to form a circle topology. The CF model can capture both the translation model of 1-D intervals as well as the rotation zones between alignment positions. This allows us to handle complex spatial reasoning queries using the CF model.

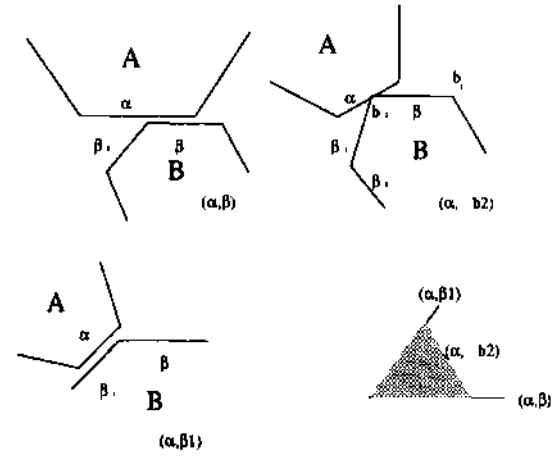

Figure 3: Rotation about a vertex as A slides further left. The middle position $\{(\alpha,b_2)\}$ represents a qualitative zone between the two alignment CFs $\{(\alpha,b_2),(\alpha,\beta),(a_2,\beta)\}$ and $\{(\alpha,b_2),(\alpha,\beta_1),(a_1,\beta_1)\}$.

To generalize to higher dimensions, we approach the configuration space models, which look at the motion space of two bodies in general. The contact formations are then discretizations on the surface of the configuration-space obstacle ([9] [12]). Next we show how any assembly motion can be expressed as a sequence of contact formations, a concept first explored in [8] in the context of Robot Grasping. Thus the chief contribution of this work is to extend the contact formation paradigm to the problem of constructing assemblies.

2 Qualitative Contacts

[Def] An *object* A is a bounded set in E^n with *boundary* δA. About a point on the boundary, any finite neighbourhood contains points both in A and not in A. A *d-face* (written as f^d), is a subset of the boundary topologically mappable onto an open disk in E^d, i.e. a d-manifold. A *d-face* is bounded by lower order hyperedges that are not contained in the face - e.g. the $(d-1)$-faces, which distinguish it from other d-faces. Typically, some geometric change is associated with the lower order faces; e.g. in polyhedra, 1-faces correspond to slope discontinuities, and constitute part of the boundary of a 2-face. For object A in E^n, δA is decomposable into manifolds of order (n-1) or less; the union of all the faces of orders 0 to (n-1) consitutes the boundary of A: $\delta A = \bigcup\{(n-1)\text{-}faces, (n-2)\text{-}faces, \ldots, 0\text{-}faces\}$.

[Def] The collection of all the faces of the object (as distinct from their union), is the *boundary-list* $\sigma A = \{(n-1)\text{-}faces, (n-2)\text{-}faces, \ldots, 0\text{-}faces\}$ e.g. in E^2 $\sigma A = \{f_i^1, f_j^0\}$, where f_i^d are the individual d-faces in A. if $\alpha_1, \alpha_2 \in \sigma A$ then $\alpha_1 \cap \alpha_2 = \phi$ and $\cup\alpha_i = \delta A$ Thus for a polygon in E^2, δA consists of all the points on its boundary, and is the union of the edges (1-faces) and the vertices (0-faces). The boundarylist σA on the other hand, is merely a list of all its 1-faces and 0-faces.

[Def] The *transformation* T_B^A traditionally viewed as mapping point ${}^B\boldsymbol{x}$ in frame B into a point ${}^A\boldsymbol{x}$ in frame A, may also be viewed as a motion that maps an object A onto a congruent object B through the same motion that would map frame A onto frame B. This transformation is a function $T_B^A = T_B^A(\vec{u})$ of the motion vector $\vec{u}$ that causes frame A to be coincident with frame B. This motion also describes the configuration of B w.r.t. A. For transformations in E^3 with six degrees of freedom, for example, $\vec{u}$ may consist of three translations and three rotations.

Configuration Space

[Def] *Configuration space* $\mathcal{C}$ of a body w.r.t. another is the space of all the configurations $\vec{u} \in \mathcal{C}$ the bodies can have w.r.t. one another. The *obstacle space* of body B w.r.t. A, $\mathcal{OS}_A(B)$, is defined as the set of motions that cause a collision between A and B. $\mathcal{OS}_A(B) = \{\vec{u} \mid \exists(x \in A)(T_B^A(\vec{u})x \in B\}$. $\delta\mathcal{OS}_A(B)$ is the boundary of this obstacle space. Figure 4 shows an example of a 2D translational configuration space for a peg-in-hole assembly task. The bodies A and B are assumed to be rigid bodies; if an assembly motion involves deformation, the configuration space approach can only model it by deforming the obstacle space accordingly. A motion path is *Contact-preserving* iff for any configuration $\vec{u}$ on the path, the interiors of the objects do not overlap, but the boundaries touch.

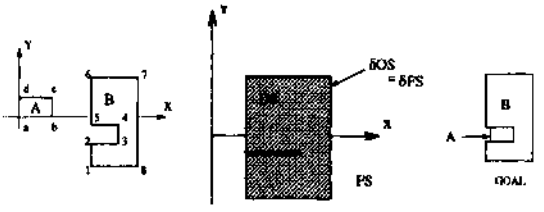

Figure 4: Configuration Space for 2D peg-in-hole assembly. The goal CF has seven elemental contacts.

Contact Formations

[Def] A *contact* λ between objects A and B is defined by the d-faces that are in contact, $\lambda = \{\alpha_i, \beta_j\}$ s.t. $\alpha_i \in \sigma A$ and $\beta_j \in \sigma B$. A contact formation A is a set of simultaneous contacts, $\Lambda_A(B)_i = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$, where λ_i are elemental contacts. If the d-faces are chosen to represent regions of qualitative continuity (such as the faces of a polyhedron), then the contact formation (CF) represents qualitatively distinguished surface contacts. For simplicity, we will represent $\Lambda_A(B)$ by A. The *contact space* $\mathcal{L}_A(B)$ is the set of all contact formations of B w.r.t. A.

We observe that the relative positions of the objects that correspond to a CF can be codified by a relative motion vector $\vec{u}$ and the corresponding rigid transformation $T(\vec{u})$. Thus there exists a mapping from the CF space onto the space of motion vectors.

3 Discretizing the Configuration Space

Lemma 1a : If $\vec{u} \in \delta\mathcal{OS}$ and $x \in$ A, $y \in$ B s.t. $x = T_B^A(\vec{u})y$, then $x \in \delta A$ and $y \in \delta B$.

Proof: Since $\vec{u}$ is on the boundary $\delta\mathcal{OS}$, any finite neighbourhood of $\vec{u}$ will include configurations both in and not in $\mathcal{OS}$, i.e. $\exists\vec{w}, |(\vec{w} - \vec{u})| < \epsilon$, and $\vec{w} \notin \mathcal{OS}$, ie. $(\forall y \in B)T(\vec{w})y \notin A$. Hence $(\forall\vec{v})|\vec{v}| < \epsilon \Rightarrow (\exists\delta > 0)\{T(\vec{u}+\vec{v})y \in Ball_\delta[T(\vec{u})y]\}$. Hence $Ball_\delta[T(\vec{u})y] - A \neq \phi$. But $T(\vec{u})B_{\vec{y}} \in A \Rightarrow T(\vec{u})B_{\vec{y}} \in \delta A \Rightarrow \vec{x} \in \delta A$. Similarly $\vec{y} \in \delta B$.

Lemma 1b : Given a configuration $\vec{u}$ in which the set of points $X \subset$ A mapped to B by the transformation $T_B^A(\vec{u})$ is on the boundary of A, i.e. $X \subset \delta A$, then the configuration $\vec{u} \in \delta\mathcal{OS}$.

Proof: Since $\vec{u}$ results in some points from A being mapped to some points in B, $\vec{u} \in (\mathcal{OS})$. If $\vec{u} \notin \delta\mathcal{OS}$, then $\vec{u}$ must be in the interior of the Obstacle Space, i.e. $\vec{u} \in int(\mathcal{OS})$. But this would imply that there is a ball around $\vec{u}$ which is contained entirely in $\mathcal{OS}$. This demands that the image of this ball will be a set of points all contained in B, and that the points interior to this ball will not be boundary points on B, which imply that the

points in A mapping to these points will also be interior points in A, violating the conditions. Hence $\vec{u} \in \delta\mathcal{OS}$.□

Lemma 2: $\forall \vec{u} \in \delta\mathcal{OS}_A(B)$, $T(\vec{u})$ represents a relative position of A w.r.t. B that represents a unique Contact Formation $A \in \mathcal{L}_A(B)$.

Proof: Let $X \subset$ A be the set of points mapped to B by $T_B^A(\vec{u})$, and let Y be its image from B. By Lemma 1, $X \subset \delta A$ and $Y \subset \delta B$. X and Y comprise of some subset of d-faces from the boundarylists σA and σB. Let any mapped pair of points (x,y) be on the faces $(f_{A_i}^q, f_{B_j}^q)$, which must be unique by the definition of Objects (no faces overlap). Hence any pair of points defines an elemental contact λ which exists and is unique. A finite number of such contacts occurring simultaneously then define a unique Contact Formation A. □

[Def] Given any Contact Formation A, the *Configuration Patch* is a set of configurations $\Gamma \subset \delta\mathcal{OS}$ such that any $\vec{u} \in \Gamma$ causes a contact in the CF A, and no $\vec{u} \notin \Gamma$ causes the CF A.

Lemma 3:

1. Any non-null CF $A \in \mathcal{L}_A(B)$ corresponds to a $\Gamma_i \subset \delta\mathcal{OS}$.
2. $\cup\Gamma_i = \delta\mathcal{OS}$.
3. $\Gamma_i \cap \Gamma_j = \phi, i \neq j$.

Proof:

1. That all CF's must have a corresponding configuration patch Γ follows from lemma 1b.

2. Let $\vec{u}$ be a configuration s.t. $\vec{u} \in \delta\mathcal{OS}$ but $\vec{u} \notin \bigcup_i \Gamma_i$.

By lemma 2, this corresponds to an unique CF A and from part 1 above, this must map to a Configuration Patch Γ. Hence if $\vec{u} \in \delta\mathcal{OS}$ then $\vec{u} \in \bigcup_i \Gamma_i$. Hence $\bigcup_i \Gamma_i = \delta\mathcal{OS}$.

3. Say $\vec{u} \in \Gamma_i$ and $\vec{u} \in \Gamma_j$. Then $\vec{u}$ corresponds to A_i and A_j which is not possible since the CF-mapping is unique [Lemma 2].

[Def] Contact Formations A_i and A_j with corresponding configuration patches Γ_i and Γ_j are *adjacent* iff $\exists$ ($\vec{u} \in \Gamma_i$ and $\vec{v} \in \Gamma_j$) s.t. a path P exists from $\vec{u}$ to $\vec{v}$ and $P \subseteq (\Gamma_i \cup \Gamma_j)$.

For example, in fig. 1, the CFs for 'overlaps' and 'equals' are adjacent, but 'equals' and 'after' are not. The transition graph (fig. 2) can be viewed as a mapping for adjacent relations in one-dimension. Note that 'after' and 'before' are not distinguished in contact formation space since both are *null* CFs. Indeed, the null CF corresponds to all configurations where there is no contact, and defines the domain for the gross motion planning problem. It may have disjoint partitions that are not reachable from one another.

Theorem: Two-body contact motions

Given a target configuration with object B in contact with A , if a path to this configuration exists from the the fully separated configuration, then such a path also exists in moving through adjacent Contact Formations.

Proof: The fully separated configuration corresponds to the null CF, and in any assembly process, there will be an initial contact, say $\vec{u}_1$. If the remaining assembly motion remains in contact, then it passes through a sequence of adjacent contact patches Γ_i's. Γ_i is adjacent to Γ_j iff the path joining any point of Γ_i to any point of Γ_j lies entirely within Γ_i and Γ_j. Since the path is connected such a sequence will always exist. Each Γ_i corresponds to some CF A_i. By definition, A_i is adjacent to A_j. If there is a contact break, the assembly returns to another null CF, and will, by the same arguments, pass through another sequence of CFs before reaching the CF containing the target configuration. Thus the path hypothesized in the theorem exists. □

This validates our claim that CFs can be used as a qualitative discretization of the motion space. In the next section we outline the algorithm for determining all CFs, and then we show how the related graph can be searched for determining solution paths to assembly problems.

4 Contact Formations for Polyhedral Objects

A direct algorithm for obtaining all the CF's would be to obtain all contact pairs *a A* x *aB,* and constrain the objects based on the contacts. If n and rn are the cardinalities of *a A* and *aB* respectively, then there will be 2^{nm} such contact pairs. This is because *nm* different kinds of elemental contacts are possible and out of these 0, 1, . . . , *nm* may occur simultaneously. However, it is clear that this is a very pessimistic bound, and while it is easy to show tighter bounds for certain classes like convex objects, a more general tighter bound appears difficult to find. For each contact, the constraint space may either be null (impossible contact), or the same as another contact (combine these elemental contacts into a contact formation). Otherwise the CF has only a single contact pair. Constraint propagation is expensive, and we present a more efficient algorithm below. The algorithm is similar to sweep models for obtaining translational C-Space, except that instead of discarding the internal structure within the C-Space we now pay particular attention to the structure there. We first identify all primary contact pairs which arise due to single feature interactions. Intersections between the primary contact pairs give rise to secondary constraints, which are essentially CFs with more than one elemental pair. The algorithm also returns a graph for the CF space in this translational slice, and the configuration space obstacle *OS* can be computed from the CF map, since it is nothing but the outer boundary of the CF map.

[Def] *Contact Formation Map* Q : A discretization on on the surface of the obstacle space of B w.r.t. A in which each region and each bounding face represents a different Contact Formation. The adjacencies between map features relate to adjacencies between their corresponding CFs.

CF Algorithm: Rotational Alignment Slices

Objective: Given reference object B and secondary ob-

ject A (A,B G E^d) in a face-alignment configuration (a rotation invariant space), find the CF map ft, i.e., all CFs that result from contact preserving motions in this slice. Let the faces in alignment be f_A and f_B.

The contact formations in this slice will involve the faces that are in contact, such as the triangular bottom face of the prismatic peg and the top face of the cube in fig. 5a. In this configuration, the two faces are parallel, but also one of the edges of the bottom face is aligned with an edge of the top face; this is not the case in fig.5c. The objective of this algorithm is to find all the CFs possible in this sliding motion. Note that the objects concerned do not have to be convex, and may even include holes.

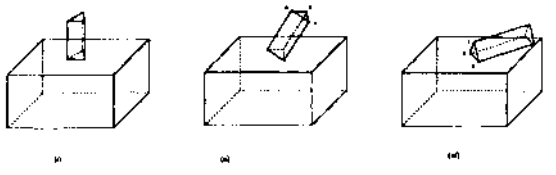

Figure 5: A prismatic peg on a cube. (i) The triangular bottom face of the peg aligned with the rectagular top face and also front edges parallel, (ii) Edge-face alignment, (iii) face-face alignment.

Contact Formation Algorithm for rotational slices

1. Record the values and ranges of all the fixed motion variables for this slice. Initialize Ω to be null.

2. Construct the object g_A by reflecting every vertex in f_A^r with respect to the frame of A, and keeping the rest of the polyhedral topology intact. g_A is also an r-face. Initialize a vertex $g^0_{A_1}$ as the reference vertex.

3. At any vertex f_B^0 on the boundary of the q-face f_B^q place a translation of the reflected face g_A so that the reference vertex $g^0_{A_1}$ is coincident with f_B^0. Add all vertices and edges of this object to Ω; e.g. the vertices are: $\Omega^0(f_B^0) = \{\omega_1^0, \omega_2^0, \ldots, \omega_k^0\} = f_B^0 \bigcup \{g_{A_2}, g_{A_3}, \ldots\}$, where the vertices g^0 are taken in the new placement.

4. At each $\omega_j^0 \in \Omega^0(f_B^0)$ translate the face f_B^q such that f_B^0 is coincident with ω_j^0. Add each face of each of these placed objects as a face in Ω.

5. For the original q-face f_B^q, at each vertex f_{Bi}^0, place a translation of the reflected face g_A such that the reference vertex $g^0_{A_1}$ coincides with each f_{Bi}^0. All faces of each of these placed objects (i.e g_A's) are added as faces in Ω.

6. The faces added to Ω so far correspond to elemental contact, arising from a single contacts. Find all intersections of these elemental contacts to form the set of Contact Formations. Add all intersection sub-faces as faces in Ω.

Figure 6 shows the result of applying this algorithm to the two faces of fig 5. The prism faces shown by dotted and dashed lines are the critical configurations where the peg has no d.o.f. available if it is to maintain the particular contact formation.

Given the CF maps for the alignment configurations, all configurations in the intermediate qualitative zones have identical topologies, which can be constructed by considering any instance of a configuration in that zone. Thus, for the middle configuration of fig 5, the CF map is obtained by sweeping the edge of the prism along the rectangle face. Non-alignment motions have no internal features and are added to ft directly.

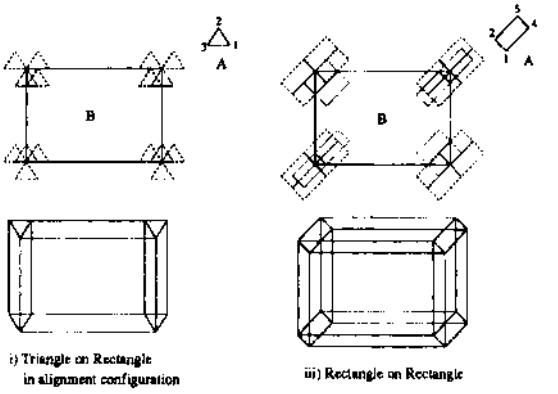

Figure 6: The CF maps for the configurations (i) and (iii) in the previous figure. The upper part shows the two faces in contact positions; the lower part the CF map. The regions correspond to the contact formations where two d.o.f.'s are available which maintain the contact, the lines where only one is available and so on.

Note that this algorithm is used to find the slices corresponding to purely translational motion such as (i) in fig 5, but also for a representative instance of a qualitatively invariant zone, such as all rotations about the alignment edge, for which the CF map in fig 6(iii) is a boundary slice.

Complexity: Let n and m be the total number of faces in A and B. The number of alignment relations depends on the number of faces that are 1-manifold or above (Vertices sliding on an object reproduce that object itself), a number that is bounded by n. In general, there are *O(nm)* candidate alignment configurations. For each of these, the algorithm above requires *0(nm)* time in computing the translated objects, and *0(k* + nrnlog(rmi)) time in evaluating the intersections for 2-faces, where k is the number of actual intersections (For 3D faces, whatever that may mean, it would be $0(n^2m^2)$). Thus, for objects upto E^3 the complexity of the algorithm for finding all CF maps for all the slices is $0(n^2m^2 \log(mn))$ ifk is less than nmlog(nra). If $n = max(n,m)$, then this is $0(n^4 \log n)$.

There are several other aspects of this algorithm. If one of the faces is non-convex, or even has a hole, the algorithm still determines the CF maps correctly, since it is based on a linear sweep between the edges, which locally results in the correct translational boundary (Fig. 7). A side benefit of this computation is that it makes it simple (an *O(nm)* planar sweep postprocessing), to compute the C-spaces for polyhedral obstacles, a problem for which solutions appear to be known only in the

convex cases [16]. A simple strategy like plane-sweep can be used to determine the boundary of the map which is then the C-space of the objects in question. Note that taking just the convex hull, as suggested earlier, will not give the C-space. A traversal through the face-list in this map, yields the CF connectivity graph as in figure 8 which can be used for assembly. In the following section, we integrate the above methodology to provide a search space for Fine Motion Planning.

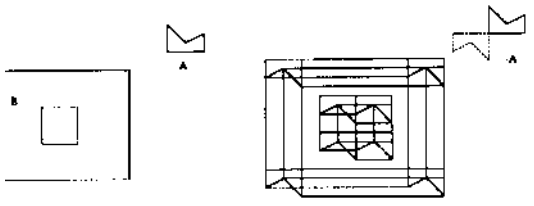

Figure 7: A CF map for a non-convex peg on a rectangle with hole.

5 The 2-Body Problem

The assembly problem consists of taking a set of bodies that were initially separated into contact. The principal difference with path planning is that there are multiple bodies (although they are often considered monotonically in subassemblies), and that the initial configuration is not fixed. Other differences include the occasional need to find space to temporarily hold subassemblies. The part of this problem that has received the maximum attention is the sequence planning problem, and the part that has received the least attention, undoubtedly because of its intrinsic difficulty, is relating it to the geometry. In the following, we present an algorithm for obtaining the contact motions of two objects, or what we call the 2-body problem. This constitutes a *single link* in an AND/OR or other "high-level" assembly graph. To our knowledge this is the first model that provides a reasonable process for mapping these connectivity links directly to the geometries of the objects. We now give an algorithm for assembly by searching through the contact formations .

Algorithm

1. Construct the contact formation map assuming full domain degrees of freedom (i.e. no limit angles etc). Also, obtain the C-space in this process.

2. Adding the constraints, consider the surfaces caused by the constraints. These surfaces do not have any internal CF features, since the entire surface corresponds to a stop position. In other words, these surfaces can delete some CF maps that arise in inaccessible regions, but cannot add any new CFs.

3. For each CF identify the configuration patch (CP) corresponding to it on the C-space. If this region is not connected (e.g. for null CF) then split the separate connected components, and label these as separate sub-CFs.

4. Scan all the CPs and connect adjacent CF's corresponding to adjacent regions on the slice by undirected edges.

This returns a CF graph indicating connections in the Contact Formation space and provides a *very basic* search space for further queries and operations. Each region in this space can be directly related to boundary features on the object. All standard AI methodologies such as learning based on densities, incremental definition etc. can be used on this graph to reduce the search complexity. It may be noted that all those regions which formed the boundary of the original C-space are part of the boundary of the constrained C-space also, except that they have added internal features on these indicating the richness of the CF structure in it.

The CF graph is useful in solving a variety of spatial problems. To answer the query about the cube rolling down a plane, originally posed in [I], we can move from slice to slice as the cube rotates, to obtain directly the sequence of vertex edge and face contacts as they are made and broken. To investigate the kinematic problem posed by Forbus et al in [IO], where one would like to say how two wheels "one with a bump on it and the other with a notch carved out of it" will travel, one can build the CF map, which provide a discretization for the Configuration Space methods proposed earlier. As for assembly, we illustrate the applicability of the CF map by outlining the solution to the peg insertion problem posed (Fig. 4). The graph resulting from the CF map is shown in Fig. 8, and the set of CFs is listed below (the edge between vertices *i* and *j* is represented by *ij).*

Another aspect of the CF graph is that each constraint-reveals the degrees of freedom that exist in maintaining that constraint, e.g in fig. 6(i), the slice is fully constrained in rotation, and hence the vertices in the graphs represent zero degrees of freedom (annotated as (CF°'s). The edges in this diagram, as well as the vertices in (iii) which permit a rotation, permit one degree of freedom and constitute CF's. The intersection process results in a depletion of freedom; the intersection of CF^r and CF^q results in a CF^S, where s - d-(d-r)-(d-q) - r+q-d. This is an important notion in designing assemblies to have certain d.o.f's and reveals where on the surface to place constraining grooves or other similar questions.

Example

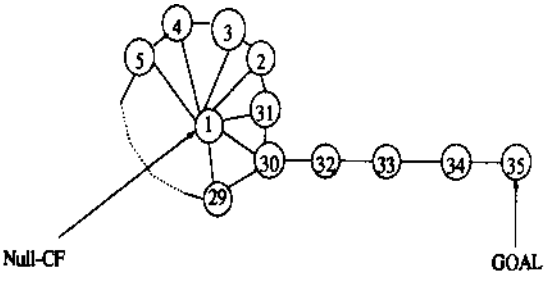

Figure 8: Contact Formation Graph. The null CF corresponds to the gross motion planning problem, and the nodes in the graph the different contact modalities

1. Null
2. $(c,2),(b,\overline{12}),(\overline{bc},\overline{12})$
3. $(c,\overline{12}),(b,\overline{12}),(\overline{bc},\overline{12})$
4. $(b,1),(c,\overline{12}),(\overline{bc},\overline{12})$
5. $(c,\overline{12}),(1,\overline{bc}),(\overline{bc},\overline{12})$
6. $(c,1)$
7. $(1,\overline{cd}),(c,\overline{18}),(\overline{18},\overline{cd})$
8. $(1,d),(c,\overline{18}),(\overline{18},\overline{cd})$
9. $(c,\overline{18}),(d,\overline{18}),(\overline{cd},\overline{18})$
10. $(c,8),(d,\overline{18}),(\overline{cd},\overline{18})$
11. $(8,\overline{cd}),(d,\overline{18}),(\overline{cd},\overline{18})$
12. $(d,8)$
13. $(d,\overline{78}),(8,\overline{ad}),(\overline{78},\overline{ad})$
14. $(8,a),(d,\overline{78}),(\overline{78},\overline{ad})$
15. $(a,\overline{78}),(d,\overline{78}),(\overline{78},\overline{ad})$
16. $(7,d),(a,\overline{78}),(\overline{78},\overline{ad})$
17. $(a,\overline{78}),(7,\overline{ad}),(\overline{78},\overline{ad})$
18. $(7,a)$
19. $(a,\overline{67}),(7,\overline{ab}),(\overline{67},\overline{ab})$
20. $(7,b),(a,\overline{67}),(\overline{67},\overline{ab})$
21. $(a,\overline{67}),(b,\overline{67}),(\overline{67},\overline{ab})$
22. $(a,6),(b,\overline{67}),(\overline{67},\overline{ab})$
23. $(6,\overline{ab}),(b,\overline{67}),(\overline{67},\overline{ab})$
24. $(b,6)$
25. $(b,\overline{56}),(5,\overline{bc}),(\overline{56},\overline{bc})$
26. $(c,6),(b,\overline{56}),(\overline{56},\overline{bc})$
27. $(b,\overline{56}),(c,\overline{56}),(\overline{56},\overline{bc})$
28. $(b,5),(c,\overline{56}),(\overline{56},\overline{bc})$
29. $(5,\overline{bc}),(c,\overline{56}),(\overline{56},\overline{bc})$
30. $(c,5),(b,2)$
31. $(b,\overline{12}),(2,\overline{bc}),(\overline{bc},\overline{12})$
32. $(b,\overline{32}),(c,\overline{54}),(\overline{cd},\overline{54}),(\overline{ab},\overline{32})$
33. $(a,2),(d,5),(b,\overline{32}),(c,\overline{54}),(\overline{cd},\overline{54}),(\overline{ab},\overline{32})$
34. $(a,\overline{32}),(b,\overline{32}),(c,\overline{54}),(d,\overline{54}),S(\overline{cd},\overline{54}),(\overline{ab},\overline{32})$
35. $(b,3),(c,4),(\overline{bc},\overline{34}),(a,\overline{32}),(d,\overline{54}),(\overline{cd},\overline{54}),(\overline{ab},\overline{32})$

We see from the figure a number of paths exist between the START and the GOAL positions eg. 1,30,32,33,34,35 and 1,3,2,31,30,32,33,34,35 etc. These correspond to alternate mechanisms for executing the assembly. Unlike other assembly planning systems, the CF map allows one to distinguish between these choices not only based on connectivity but also on other functional issues such as landmark configurations, the existence of similar error configurations, etc.

6 Conclusion

This work provides a novel methodology for spatial objects in 2D and 3D, to derive qualitative maps of relative motions. The discretization claims to be qualitative since it is compact, the entire set is determinable based on functional features of the domain, and the knowledge maintained provides direct explanation power since the set of surface features in contact are critical to explaining many tasks. As illustrated in our focus domain involving parts assembly, the applications of this procedure can provide rich dividends in a number of problem areas. In a sense this work is a synthesis of radically different lines of thought from robotics, qualitative reasoning, assembly planning, and vision, but the end result sufficiently different from the methodologies in any of these to be viewed as a new tool in the arsenal of qualitative reasoning.

References

[1] Varol Akman and Paul J. W. ten Hagen. The power of physical representations. *AI Magazine*, pages 49-65, 1989.

[2] James F. Allen. Maintaining knowledge about temporal intervals. *IEEE PAMI*, Vol. 26(11):832-843, November 1983.

[3] B. Bhatia. *Automatic Robotic Assembly from Disassembly* sis, Indian Institute of Technology, Kanpur, India, Center for Robotics, Mechanical Engineering, Oct. 1992.

[4] A. Bourjault. *Contribution a une Approche Methodolodique de L'Assemlage Automatize: Elaboration Automatique des Sequence Operatores*. PhD thesis, Universite de Franche-Comte, Besancon, France, Nov. 1984.

[5] J. F. Canny. *The Complexity of Robot Motion Planning*. The MIT Press, Cambridge, 1987.

[6] Eliseo Clementini and Paolino Di Felice. A comparison of methods for representing topological relationships. *Information Sciences*, to appear.

[7] L. S. Homem de Mello and A. C. Sanderson. And/or graph representation of all assembly sequences. *IEEE Trans. of Robotics and Automation*, Vol. (6)(2):188-199, Apr. 1990.

[8] Rajiv Desai and Richard A. Volz. Identification and verification of termination conditions in fine motion in presence of sensor errors and geometric uncertainties. *IEEE Conference of Robotics and Automation*, May 1989.

[9] Boi Faltings. Qualitative kinematics in mechanisms. *Artificial Intelligence*, Vol. 44(1-2):89-119, 1990.

[10] Kenneth D. Forbus, Paul Nielsen, and Boi Faltings. Qualitative spatial reasoning: the clock project. *Artificial Intelligence*, Vol. 51:417-471, October 1991.

[11] Ziv Gigus, J. Canny, and R. Seidel. Efficiently computing and representing aspect graphs of polyhedral objects. *CACM*, Vol. 13(6):542-551, 1991.

[12] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Massachussets USA, 1991.

[13] S. Lee. Backward assembly planning with assembly cost analysis. In *IEEE Conference of Robotics and Automation*, pages 2383-2391, 1992.

[14] Amitabha Mukerjee and G. Joe. A qualitative spatial representation based on tangency and alignment. Technical Report 92-003, Texas A&M University CS Dept, Feb 1992. Shorter version in AAAI-90 (p.721-727).

[15] Amitabha Mukerjee and Nishant Mittal. A qualitative representation of frame-transformations in 3-dimensional space. Technical Report ME-94-02, IIT Kanpur Department of Mechanical Engg, 1994.

[16] T. Lozano Perez. Spatial planning: A configuration space approach. *IEEE Trans. Computers*, Vol. (C-32)(2):108-120, Feb. 1983.

[17] David Pierce and Benjamin Kuipers. Learning to explore and build maps. *AAAI-94*, pages 1264-1271, 1994.

[18] J. Reif. Complexity of the mover's problem and generalizations. In *Proc. 20th IEEE Symposium FOCS*, 1979. Also in *Planning, Geometry and ,Complexity of Robot Motion*, edited by J. Schwartz, J. Hopcroft and M. Sharir, Ablex publishing corporation, New Jersey, 1987, Ch. 11, pages 267-281.

[19] Randall H. Wilson and Jean-Claude Latombe. Geometric reasoning about mechanical assembly. *Artificial Intelligence*, v.72:1-2:371-396, November 1994.

[20] J. Wolter, S. Chakrabarty, and J. Tsao. Mating constriant languages for assembly sequencing planning. *IEEE Conference of Robotics and Automation*, pages 2367-2374, May 1992.