ASSEMBLY FEATURE-MATING INFERENCE FROM SOLID MODELS USING SYMMETRY GROUPS

Yanxi Liu and
Robin J. Popplestone

Computer and Information Science Department
University of Massachusetts

COINS Technical Report 90-34

Assembly Feature-mating Inference from Solid Models Using Symmetry Groups[1]

Yanxi Liu
Robin J. Popplestone

Laboratory for Perceptual Robotics
Department of Computer and Information Science
University of Massachusetts at Amherst, USA

ABSTRACT

This chapter describes how to use geometric boundary models of assembly components to find mating features, thereby permitting simpler task specifications to be used for robot assembly planning. This forms part of our work on a high-level robot assembly task planner. To mitigate the combinatorics of searching for a feasible correspondence between features of different bodies, boundary models produced by a geometric modeller PADL2 are matched against a library of standard compound features. The feasibility of mating compound features and the relative position of assembly components are analyzed using the symmetry groups of features. A formal treatment of features and their symmetry groups is described.

[1]This paper is to appear as a book chapter in "Progress in Robotics and Intelligent Systems", C.Y. Ho and G. Zobrist (Eds.), Ablex Publishing Corporation, New Jersey, 1990.

Contents

List of Figures

List of Tables

1 Introduction

For assembly purposes, robots are built to be general purpose flexible machines which can perform a variety of different assembly tasks without costly changes to hardware and software. However, even if we had such a perfect general purpose machine it would not be very useful if it were hard to instruct. An analogy can be made here between a robot and a 'naked' computer, just imagine how difficult it would be for human beings if the only language to communicate with the computer was the machine language for that specific machine. Naturally, task-level planning in robotics has drawn the attention of researchers [19].

In robotic task-level planning [14, 10], it is assumed that a task specification expressed in terms of objects to be handled by a robot is given, the task-level planner then translates such a specification into a plan expressed in terms of actions by the robot. From a mechanical design it is not always trivial to give a complete and unambiguous assembly task specification. For some simple examples, such as the one shown in Figure 1, an assembly configuration specification could be tedious due to *symmetries* of the assembly components. It is therefore desirable to provide a means for inferring the details from a simpler assembly task specification, or in another words, to enhance the knowledge of a robot planner such that it can understand symmetry. In our work [11, 23] this intention has been pursued through reasoning about symmetry properties of assembly components automatically by the planning system. Although the symmetry of an assembly component initially appears to be a problem, it actually plays a very helpful role in generating assembly plans.

Bodies in an assembly mate through their features. A *primitive feature* F is a surface which is a subset of the Euclidean space $\Re^3$ and is associated with a boundary model of a solid. Examples are infinite planes, cylinders and spheres. A *compound feature* is a set of primitive features. This usage corresponds essentially to that of Woodbury [27], and captures the idea of features used by such workers as Dixon [6], Henderson [9] , Green, Carney and Brown [7, 3]. In specifying assemblies it is possible that the actual mating *features* of bodies are omitted — only the fact that the bodies themselves are to be mated is stated. The sets of possible mating features of two bodies are inferred from their geometric models. This distinguishes our work from most of the robot task-level assembly planning systems where a detailed, unambiguous assembly task specification is required.

The geometry of the assembly components enforces some *hard constraints* on the configuration(s) of assembly and the order of assembly. We have used for assembly planning the geometric modeller PADL2 [2, 24]. This modeller is capable of representing certain algebraic surfaces with a precision limited only by floating point accuracy. While workers such as Lozano-Pérez et al [15] have used polyhedral modellers for significant studies in the integration of recognition and manipulation, for planning the manipulation of real engineered objects it is important to have representations in which curved surfaces are represented as such. While other workers have used special purpose, user-defined representations for assembly components, such as the relational model in [25], there is relatively little use made of non-polyhedral models that are geometrically complete.

From a library of *compound features* the planner finds semantically relevant features in the set of geometric entities which constitute the boundary model of an object. Such semantically relevant features then are used as the basis of a heuristic search for mating features.

Any subset S of $\Re^3$ may have *symmetries.* A symmetry is a rotation and/or translation which leaves S setwise invariant. For example, a rotation about the central axis of a cylinder is a symmetry of the cylinder. Although some points of the cylinder are relocated by the rotation, the cylinder as a whole still occupies the same volume in $\Re^3$. Features may have symmetries as well. Such symmetries of features in an assembly are of essential importance in understanding how features mate and what the final assembly configurations can be.

All the possible rigid motions in 3-space form a *group* called the Proper Euclidean group $\mathcal{E}^+$. We can regard a group simply as a set of mappings which satisfy certain additional *closure* properties, as described in section 4. The set of all the symmetries of a feature has a group structure. This group is called the *symmetry group* of the feature. Any symmetry group is a subgroup of $\mathcal{E}^+$. In assembly planning the symmetry of features is usually more important than the symmetry of the bodies to which the features belong because bodies are mated through their features. For example, consider a cylindrical hole in a block. This feature is mapped into itself by any rotation about the axis of the hole: such rotations are therefore symmetries of the hole but not in general symmetries of the block.

The *symmetry group* is an important feature descriptor in our work and has been integrated with the geometric model of each assembly body feature. Combining n features into one compound feature F_{comp} requires the calculation of the symmetry group of F_{comp}, to which the method of treating the intersection of symmetry groups developed in [13] has been applied.

The work described here shows that an assembly planning system equipped with a feature library and the ability to represent and reason about symmetries can 'understand' a much broader and simpler task specification than one without. Thus a higher level of automation is achieved. Furthermore, a precise and relevant task specification can be generated for lower-level robotic planning.

The next two sections detail the input and output of PADL2, a geometric solid modeller; Section 4 is devoted to a more formal treatment of features and their symmetry groups. Section 5 discusses how to recognize relevant compound features in an assembly, how to match boundary models against the feature patterns of a feature library, and how to compute the symmetry group of a compound feature. Section 6 states the necessary and sufficient conditions for a pair of features to mate. Section 7 explains how the planner works through a detailed example and the trace of a program execution. In the concluding section we assess the significance of this work in assembly planning.

2 How to Describe Shapes to PADL2

We use Prolog terms to express Constructive Solid Geometry as described in [21]. This formalism is translated into the input syntax for PADL2 by the Prolog predicate *draw*. All of the involved assembly parts are thus treated. For bodies *b*1 and *b*4 in Figure 1, the requests to PADL2 are :

```
;;;b1
:- set_solid(b1, (cyl(4,1)    \/  (cyl(4,1)@trans(4,0,0))     \/
                 (block(1,3,7)@trans(-1.5,-1.5,1)@rot(jj,1.570796)))).

:-draw(b1).

;;;b4
:- set_solid(base,
        ((((( (block(15,12,2) \ cyl(2,1)@trans(5,2,1) )  ;;; 8
               \ cyl(2,1)@trans(9,2,1) )                 ;;; 9
               \ cyl(2,1)@trans(4,10,1) )                ;;; 10
               \ cyl(2,1)@trans(12,5,1) )                ;;; 11
```

```
                \ cyl(2,1)@trans(12,9,1) )                  ;;; 12
                \ cyl(2,2)@trans(8,6,1)                     ;;; 13
            ) @trans(0,0,10) ) ).
:-draw(base).
```

Here \/ means set union, / means set difference and @ is used to relocate bodies by transformations. The forms in *cyl* and *block* are primitives of CSG description. The Prolog predicate *draw* tells PADL2 to form and display a boundary model of the object.

PADL2 provides only a limited surface description capability. For example you cannot define tapped holes in PADL2, nor the exact form of involute gearing, and even if PADL2 provided this capability, such detailed modelling would be very expensive in time and space. A way around this difficulty is to regard such surface forms as texture. Thus we have extended the CSG formalism to include primitives which have these surface forms, e.g. *threaded_cyl*(l, r, p, P), where p is a pitch and P is a profile. This is passed to PADL2 just as a cylinder, but the surfaces in the extracted boundary model are re-labeled, and the symmetry group is the appropriate screw group (Table 1).

3 How PADL2 represents solids— — —boundary models of solids

The internal representation of the boundary model produced by PADL2 is extracted using FORTRAN subroutines linked into POPLOG [8] as external procedures; POP-11 record structures expressing the face-edge-vertex structure of a body are built from the extracted information. A boundary representation which contains the surfaces, lines and vertices and their locations can then be saved and restored as a file. Each solid (one individual assembly part) is represented as :

```
recordclass solid
    assem_of_solid    ;;;parent assembly record
    faces_of_solid    ;;;list of face records
    edges_of_solid    ;;;list of edge records
    vertices_of_solid ;;;list of vertex records
```

Table 1: Correspondence between shape and its symmetry group

Subset $S \subset \Re^3$	surface name	Symmetry group
$\mathbf{H}$	half plane	$\mathcal{G}_{plane}$
$\mathbf{Cyl(r)}$	cylinder	$\mathcal{G}_{cyl}$
$\mathbf{Sph(r)}$	sphere	$SO(3)$
$\mathbf{Screw(p, r)}$	screw	$\mathcal{G}_{screw}(p)$
$\mathbf{Gear(p_r, p_a, n)}$	gear	D_{2n}
$\mathbf{Cone(\theta)}$	cone	$SO(2)$

Each face contains the following information:

```
recordclass face
    face_id           ;;;integer used as feature name
    type_of_face      ;;;padl2 mnemonic
    surface_of_face   ;;;3-d geometric entity
    sign_of_face      ;;;integer -1 or 1
    solid_of_face     ;;;parent solid record
    edges_of_face     ;;;list of edge records
    vertices_of_face  ;;;list of vertex records
    adj_faces_of_face ;;;list of adjacent face records
    loc_of_face       ;;;location matrix (4X4 matrix)
    sym_of_face       ;;;canonical symmetry Group of the face
```

After extraction of the PADL2 boundary model, surface types of faces are mapped to their canonical symmetry group elements [22]. Table 1 gives some correspondences of surface type and its symmetry group. The reason that those infinitely extended surfaces and their symmetry groups on Table 1 are important and useful is that finite solids are bounded by such surfaces. We shall discuss features and their symmetry groups in detail in Section 4.

Each *feature* recordclass contains:

```
recordclass feature
    feature_name            ;;;  The name of the feature
```

```
feature_type              ;;; whether it is a compound feature
feature_group             ;;; The symmetry group of the feature
feature_geo_params        ;;; The geometric parameters
feature_face;             ;;; list of face recordclass
```

The symmetry group recordclass contains:

```
recordclass gr
    gr_canon         ;;; the name of the canonical symmetry group
    gr_loc           ;;; the 4X4 transformation matrix
    gr_trans_inv     ;;; the translational invariant
    gr_poles;        ;;; the set of poles
```

An example of a primitive feature with a cylindrical surface is as follows:

```
<feature 8
        CSURF
        <gr gr_dir_cyl
         1.0      0.0      0.0      5.0
         0.0      1.0      0.0      2.0
         0.0      0.0      1.0      11.0
         0        0        0        1
         <ln3 0.0 0.0 0.0 0.0 0.0 1.0>
         [<pole <pt3 0 0 1> 0> <pole <pt3 0 0 -1> 0>]
         >
        [-1 <cyl3 1.0 <ln3 5.0 2.0 11.0 0.0 0.0 1.0>>]
        [Face 8 CSURF

  surface:  <cyl3 1.0 <ln3 5.0 2.0 11.0 0.0 0.0 1.0>> sign:  -1
  texture:  tex_smooth  symmetry:  gr_dir_cyl

         1.0      0.0      0.0      5.0
         0.0      1.0      0.0      2.0
         0.0      0.0      1.0      11.0
         0        0        0        1
```

```
    edges:    5 18

    vertices:

    faces:    7 3

]>
```

Here the $<$ and $>$ angle brackets enclose a labeled tuple which is a POP-11 record. Thus the *feature_name* of the surface, given by PADL2, is 8. Its *feature_type* is "CSURF", the PADL2 mnemonic for a cylindrical surface. Its symmetry group is the *directed cylinder* group *gr_dir_cyl* and it is re-located by the above 4×4 matrix. The axis of symmetry for the feature is the line which passes through the point $(5.0, 2.0, 11.0)$ and is parallel to the Z-axis. The radius of the cylindrical surface is 1 unit. That it is a concave surface is indicated by the integer -1, the sign of the surface.

4 Features and their Symmetry Groups

In this section we shall start with some useful definitions for abstract groups and, in particular, the Euclidean group. Then we introduce the concept of a feature and its symmetry group. For some basic topological definitions see Section 9.

4.1 Abstract groups, the Euclidean group $\mathcal{E}^+$ and subgroups of $\mathcal{E}^+$

Definition 1 *An abstract* **group** *G is a set of elements closed under an associative composition operation which we shall write as multiplication, with an identity element $1 \in G$, and an inverse for any $g \in G$ which obeys the laws:*

$$g1 = g = 1g, \; gg^{-1} = 1 = g^{-1}g$$

A **subgroup** H of G is a subset of G that is itself a group. An associated concept is a **coset** of a subgroup H of G. For any element g in G, $gH = \{gh | h \in H\}$ is a

left coset and Hg is a **right coset** of H in G. Accordingly if $g_1, g_2 \in G$ then $g_1 H g_2$ is called a **two-sided coset** of H in G. A coset is not in general a group.

Let R^3 be three-dimensional real Euclidean space. As is well known, the bilinear form

$$< x, y >= \sum_{i=1}^{3} x_i y_i, \quad x, y \in R^3$$

defines an inner product on this space. The norm $||x|| =< x, x >^{1/2}$ is the **Euclidean length** of $x \in R^3$.

Definition 2 Isometries *A map* $R^3 \xrightarrow{g} R^3$ *is an* **isometry** *if and only if* $\forall x, y \in R^3$ $||g(x) - g(y)|| = ||x - y||$.

Definition 3 The Euclidean Group *The set of all the isometries of R^3 forms a group with the composition of isometries as group product; this group is called the* **Euclidean group**. *Its elements are rotations, translations, reflections and combinations of these.*

Definition 4 The Proper Euclidean Group

If we restrict the set of isometries to those isometries which are rigid transformations, thus excluding reflections, then the remaining set still has a group structure and is called the Proper Euclidean group, *denoted by* $\mathcal{E}^+$.

The theory of robotic assembly is the theory of how rigid motions imposed by robots act on $3D$ bodies, subsets of the Euclidean space, to realize certain spatial relationships among these bodies. Therefore **the proper Euclidean group** $\mathcal{E}^+$ and its subgroups are particularly important.

4.2 Assembly features and their symmetry groups

In this section we give a formal definition of the symmetries of a feature, and then immediately show that they form a group, called the symmetry group of the feature. This provides us with the basic justification for the use of group theory in assembly planning.

Definition 5 *A* **primitive feature** *F is a connected subset of $\Re^3$ that is a smooth, irreducible algebraic surface.*

Examples include surfaces of infinite plane, infinite cylinder, cone, sphere and torus. A specific PADL2 model contains data-structures specifying such primitive features.

Definition 6 *Let* $S \subseteq \Re^3$*. Then* $g \in \mathcal{E}^+$ *is a* proper symmetry *of* S *if and only if* $g(S) = S$.

Proposition 1 *The proper symmetries of a set* $S \subseteq \Re^3$ *form a subgroup of* $\mathcal{E}^+$.

Proof:

Let G denote the set of the symmetries of $S \subset \Re^3$. Obviously, $1(S) = S$, so $1 \in G$. If $g \in G$ then $g(S) = S$, multiplying by g^{-1} we have $g^{-1}g(S) = g^{-1}(S)$ therefore $g^{-1}(S) = S$ and so $g^{-1} \in G$. Finally, if $g_1, g_2 \in G$ then $(g_1g_2)(S) = g_1(g_2(S)) = g_1(S) = S$ therefore $g_1g_2 \in G$. By the definition of a subgroup (Definition 1) G is a subgroup of $\mathcal{E}^+$. □

Definition 7 *When* S *is a feature, the above group* G *associated with* S *is called the* **symmetry group** *of the feature* S.

Definition 8 *Two primitive features* F_1, F_2 *are said to be*

- **distinct** if for all the open subsets[2] $f_1 \subset F_1, f_2 \subset F_2$ no such $g \in \mathcal{E}^+$ exists such that $g(f_1) = f_2$. See Figure 2 for an example of two distinct features.
- **1-congruent (weakly-congruent)** if there exists at least one $g \in \mathcal{E}^+$ such that $g(F_1) = F_2$, but for all such $g \in \mathcal{E}^+, g(F_2) \neq F_1$. For an example see Figure 3.
- **2-congruent (strongly-congruent)** if there exists $g \in \mathcal{E}^+$ such that $g(F_1) = F_2$ and $g(F_2) = F_1$. For an example see Figure 4.

It is important to introduce the concept of *conjugate groups* to understand how features and their symmetry groups are related.

Definition 9 *The two subgroups* $H_1, H_2 \subset G$ *are said to be* **conjugate** *if* $H_2 = aH_1a^{-1}$ *for some* $a \in G$.

[2]They are open with respect to the induced topology from $\Re^3$

Proposition 2 *If G is the symmetry group of $S \subset \Re^3$ then for any rigid transformation a in $\mathcal{E}^+, aGa^{-1}$ is the symmetry group of $a(S)$.*

Proof:

Let $H = aGa^{-1}$ and let H_a be the symmetry group of $a(S) \subset \Re^3$. If $h \in H$ then there exists a $g \in G$ such that $h = aga^{-1}$, and moreover $g(S) = S$. Then $h(a(S)) = aga^{-1}(a(S)) = ag(S) = a(S)$. Thus h is a symmetry of $a(S)$, and so $h \in H_a$. Thus we can conclude $H \subseteq H_a$.

Conversely, if $h_a \in H_a$ then $h_a(a(S)) = a(S)$ and so $a^{-1}h_a a(S) = S$, i.e. it is a symmetry of S. Thus $a^{-1}h_a a = g \in G$ and $h_a = aga^{-1} \in H$ therefore $H_a \subseteq H$.

Thus we conclude $H = H_a$. □

By Proposition 2, when a feature is relocated by a transformation g, the symmetry group of the relocated feature will be the conjugation by g of the symmetry group of the original feature. One approach to representing any feature symmetry group is to make it be a conjugate of *a canonical symmetry group.* These canonical groups are chosen in a systematic way — if they have a single axis of rotation it is chosen to be the Z-axis, if they leave a single point of 3-space fixed, that point is chosen to be the origin, etc.

A symmetry group of S can be represented by a pair consisting of a *canonical symmetry group* G_{canon} and an element g of $\mathcal{E}^+$ which transforms S from the origin to its current location. Table 1 gives some of the correspondences between subsets of R^3 and their *canonical* symmetry groups. A list of some important canonical subgroups of $\mathcal{E}^+$ with their definitions is given in Table 2. Figure 6 shows subgroup relationships between some important subgroups of $\mathcal{E}^+$. The arrow $\mathcal{G}_1 \rightarrow \mathcal{G}_2$ in Figure 6 means that $\mathcal{G}_2$ is a subgroup of $\mathcal{G}_1$.

4.3 Compound features and their symmetry groups

Now let us formally define a compound feature and its symmetry group. We are also going to prove that under certain conditions the symmetry group of a compound feature is the intersection of the symmetry groups of its component primitive features.

Definition 10 *A* **compound feature** *is a set union of some primitive features.*

Table 2: Some important subgroups of $\mathcal{E}^+$

Canonical Groups	Definition
$\mathcal{G}_{id}$	$\{1\}$
$\mathcal{T}^1$	$\mathbf{gp}\{\mathbf{trans}(0,0,z) \| z \in \Re\}$
$\mathcal{T}^2$	$\mathbf{gp}\{\mathbf{trans}(x,y,0) \| x,y \in \Re\}$
$\mathcal{T}^3$	$\mathbf{gp}\{\mathbf{trans}(x,y,z) \| x,y,z \in \Re\}$
$SO(3)$	$\mathbf{gp}\{\mathbf{rot}(\mathbf{i},\theta)\mathbf{rot}(\mathbf{j},\sigma)\mathbf{rot}(\mathbf{k},\phi) \| \theta,\sigma,\phi \in \Re\}$
$SO(2)$	$\mathbf{gp}\{\mathbf{rot}(\mathbf{k},\theta) \| \theta \in \Re\}$
$O(2)$	$\mathbf{gp}\{\mathbf{rot}(\mathbf{k},\theta)\mathbf{rot}(\mathbf{i},n\pi) \| \theta \in \Re, n \in \mathcal{N}\}$
$\mathcal{G}_{cyl}$	$\mathbf{gp}\{\mathbf{trans}(0,0,z)\mathbf{rot}(\mathbf{k},\theta)\mathbf{rot}(\mathbf{i},n\pi) \| n \in \mathcal{N}, \theta, z \in \Re\}$
$\mathcal{G}_{dir\_cyl}$	$\mathbf{gp}\{\mathbf{trans}(0,0,z)\mathbf{rot}(\mathbf{k},\theta) \| z,\theta \in \Re\}$
$\mathcal{G}_{plane}$	$\mathbf{gp}\{\mathbf{trans}(x,y,0)\mathbf{rot}(\mathbf{k},\theta) \| x,y,\theta \in \Re\}$
$\mathcal{G}_{screw}(p)$	$\mathbf{gp}\{\mathbf{trans}(0,0,z)\mathbf{rot}(\mathbf{k},2z\pi/p) \| z \in \Re\}$
$\mathcal{G}_{T_1C_2}$	$\mathbf{gp}\{\mathbf{trans}(0,0,z)\mathbf{rot}(\mathbf{i},n\pi) \| n \in \mathcal{N}, z \in \Re\}$
D_{2n}	$\mathbf{gp}\{\mathbf{rot}(\mathbf{k},2\pi/n)\mathbf{rot}(\mathbf{i},m\pi) \| m,n \in \mathcal{N}\}$
C_n	$\mathbf{gp}\{\mathbf{rot}(\mathbf{k},2\pi/n) \| n \in \mathcal{N}\}$
$\mathcal{E}^+$	$\mathbf{gp}\{\mathbf{trans}(x,y,z)\mathbf{rot}(\mathbf{i},\theta)\mathbf{rot}(\mathbf{j},\sigma)\mathbf{rot}(\mathbf{k},\phi) \| x,y,z,\theta,\sigma,\phi \in \Re\}$

Proposition 3 *Let $F_1, ..., F_n$ be pairwise distinct primitive features with symmetry groups $G_1, ...G_n$ respectively and $F = F_1 \cup ... \cup F_n$ be a compound feature with symmetry group G. Then $G = G_1 \cap ... \cap G_n$.*

Proof:
Let $g \in G$, then $g(F) = F$. Thus $g(F_1 \cup ... \cup F_n) = g(F_1) \cup ... \cup g(F_n) = F_1 \cup ... \cup F_n$. Suppose $g(F_i) \not\subseteq F_i$ then there exists a point $x \in g(F_i)$ and for some $j \neq i, x \in F_j - F_i$.

Now let ϵ be the distance between x and F_i, since F_i is closed $\epsilon > 0$. Since $x \in g(F_i)$, there must exist $y \in F_i$ such that $x = g(y)$. Since all the $F_1, ..., F_n$ are closed and pairwise distinct, their intersections have a lower dimension (edges) which do not cover any patches of any primitive feature completely. Thus there must exist a point y_1 with $||y_1 - y|| < \epsilon/2$ and $y_1 \in F_i - (F_1 \cup ...F_{i-1} \cup F_{i+1} ... \cup F_n)$. Let f_i be a neighborhood of y_1 that is completely contained in $F_i - (F_1 \cup ...F_{i-1} \cup F_{i+1} ... \cup F_n)$ with radius less or equal to $\epsilon/2$. Since g is an isometry (a distance preserving map), every point of f_i must be mapped by g to another point within distance ϵ to the point $g(y) = x \in F_j$, thus $F_i \cap g(f_i) = \emptyset$. Since the connectivity and the neighborhoodness of f_i must be preserved by the isometry g, $g(f_i)$ cannot intersect with more than one primitive feature because otherwise $g^{-1}(g(f_i)) = f_i$ has to be in the intersection of at least two features. Thus we have $g(f_i) = f \subset F_k$, for some $k \neq i$. Then $g^{-1}(f) = f_i \subset F_i \Rightarrow F_i, F_k$ are not distinct, a contradiction. Therefore $g(F_i) \subseteq F_i$. Since g is a bijection, $g(F_i) = F_i \Rightarrow g \in G_i$ for $i = 1, ..., n$. Thus $g \in G_1 \cap ... \cap G_n \Rightarrow G \subseteq G_1 \cap ... \cap G_n$.

For all $g \in G_1 \cap ... \cap G_n, g(F) = g(F_1 \cup ... \cup F_n) = g(F_1) \cup ... \cup g(F_n) = F_1 \cup ... \cup F_n = F \Rightarrow g \in G \Rightarrow G_1 \cap ... \cap G_n \subseteq G$.

Therefore $G = G_1 \cap ... \cap G_n$. □

Lemma 1 *Let a compound feature $F = F_1 \cup F_2$ have symmetry group G, where F_1, F_2 are primitive features with symmetry groups G_1, G_2 respectively, and F_1, F_2 are separated and weakly-congruent. Then for all $g \in G, g(F_1) = F_1$ and $g(F_2) = F_2$.*

Proof:
For all $g \in G, g(F) = F$, i.e. $g(F_1 \cup F_2) = g(F_1) \cup g(F_2) = F_1 \cup F_2$. $g(F_1)$ is a connected subset of $F_1 \cup F_2$ (Theorem 2). By Theorem 3 (see Section 9), $g(F_1) \subseteq F_1$ or $g(F_1) \subseteq F_2$. Because F_2 is connected $g(F_1) \subseteq F_1 \Rightarrow g(F_1) = F_1$

and $g(F_1) \subseteq F_2 \Rightarrow g(F_1) = F_2$. However if $g(F_1) = F_2$ then $g(F_2) = F_1$. That is to say that F_1, F_2 are strongly congruent by definition, a contradiction. Therefore $g(F_1) = F_1$ and $g(F_2) = F_2$. □

Proposition 4 *Let a compound feature $F = F_1 \cup F_2$ have symmetry group G, where F_1, F_2 are primitive features with symmetry groups G_1, G_2 respectively, and F_1, F_2 are separated and 1-congruent. Then $G = G_1 \cap G_2$.*

Proof:

By Lemma 1, for all $g \in G, g(F_1) = F_1, g(F_2) = F_2$. Then $g \in G_1 \cap G_2$. So we have $G \subseteq G_1 \cap G_2$. For all $g \in G_1 \cap G_2$, $g(F) = g(F_1 \cup F_2) = g(F_1) \cup g(F_2) = F_1 \cup F_2 = F$. Therefore $g \in G \Rightarrow G_1 \cap G_2 \subseteq G$. We conclude $G = G_1 \cap G_2$. □

Lemma 2 *Let a compound feature $F = F_1 \cup F_2$ have symmetry group G, where F_1, F_2 are primitive features with symmetry groups G_1, G_2 respectively, and F_1, F_2 are separated and* 2-congruent *by $g \in \mathcal{E}^+$. Then for all $g \in G$, either $g(F_1) = F_1$ and $g(F_2) = F_2$ or $g(F_1) = F_2$ and $g(F_2) = F_1$.*

Proof: For all $g \in G, g(F) = F$, i.e. $g(F_1 \cup F_2) = g(F_1) \cup g(F_2) = F_1 \cup F_2$. By Theorem 3 (Section 9), $g(F_1) \subseteq F_1$ or $g(F_1) \subseteq F_2$. Because of the connectivity of F_1 and F_2, if $g(F_1) \subseteq F_1$ then $g(F_1) = F_1$ and $g(F_2) = F_2$; if $g(F_1) \subseteq F_2$ then $g(F_1) = F_2$ and $g(F_2) = F_1$. □

Proposition 5 *Let a compound feature $F = F_1 \cup F_2$ have symmetry group G, where F_1, F_2 are primitive features with symmetry groups G_1, G_2 respectively, and F_1, F_2 are separated and 2-congruent by $g_c \in \mathcal{E}^+$ i.e. $g_c(F_1) = F_2$ and $g_c(F_2) = F_1$. Then $G =< g_c > (G_1 \cap G_2)$ where $< g_c >$ denotes the smallest group generated from g_c.*

Proof:

If $g \in G$ then by Lemma 2 either $g(F_1) = F_1$ and $g(F_2) = F_2$ then $g \in G_1$ and $g \in G_2 \Rightarrow g \in G_1 \cap G_2$; or $g(F_1) = F_2$ and $g(F_2) = F_1$ then g can be written as $g = g_c g_c^{-1} g$. Let $g_0 = g_c^{-1} g$ then $g(F_1) = g_c g_0(F_1) = F_2 \Rightarrow g_0(F_1) = g_c^{-1}(F_2) = F_1$ therefore $g_0 \in G_1$ and similarly, $g_0 \in G_2$. Thus $g_0 \in G_1 \cap G_2$. Then $g = g_c g_0 \in < g_c > (G_1 \cap G_2)$.

If $g \in < g_c > (G_1 \cap G_2)$ then $g = g' g_{12}$ where $g' \in < g_c >$ and $g_{12} \in G_1 \cap G_2$. Then $g(F) = g(F_1 \cup F_2) = g(F_1) \cup g(F_2) = g' g_{12}(F_1) \cup g' g_{12}(F_2) = g'(F_1) \cup g'(F_2)$. Since $< g_c >$ is generated from g_c, for all the members g' in $< g_c >$ either $g'(F_1) \cup g'(F_2) = F_1 \cup F_2 = F$ or $g'(F_1) \cup g'(F_2) = F_2 \cup F_1 = F$. In either case $g \in G \Rightarrow < g_c > (G_1 \cap G_2) \subseteq G$. Thus we conclude $G =< g_c > (G_1 \cap G_2)$. □

5 Compound Feature Recognition and Computation

Given n primitive features on a body, the number of *all* the possible compound features, i.e. all the non-empty subsets of a set of order n, is: $\sum_{i=1}^{n} C_n^i = 2^n - 1$. For a pair of bodies, each having m and n primitive features respectively with $m \leq n$, the number of possible mating feature pairs between the two bodies would be: $\sum_{i=1}^{m} C_n^i C_m^i$. This shows that the size of the solution space is exponential in the number of the surfaces of a body. Given the combinatorics of this problem, we have explored one approach which is to identify relevant compound features of a body by matching against feature patterns appearing in a pre-established library. After a compound feature is recognized, the symmetry group for this compound feature is computed by intersecting the symmetry groups of its primitive features.

5.1 Compound feature recognition: compound feature library

The feature library was built based on a survey [18] of some of the typical features in assembly together with some additional general feature types we found useful. Some of the library features are quite specific such as: *countersink, counterbore, keyway* and certain cases of *spline.* More generic assembly-relevant features are *insertors, containers, multi-insertors* and *multi-containers*, which are in effect general protrusions, concavities, and combinations of these. These definitions refer to the faces of the features of a single body and the relationships between them, such as being adjacent, perpendicular, parallel etc.. The library was implemented in POPLOG PROLOG, in which the *is* predicate evaluates *any* POP-11 procedure that returns one result. A sample of the feature description language is shown in the following example for one of the *insertor* types, which corresponds to the end of a cylinder:

```
find_an_insertor(B,I) :-          ;;; B is a name of a body
                                  ;;; recordclass
  A is valof(B),                  ;;; get the value of B
  Face is faces_of_solid(A),      ;;; get a list of faces of A
  member(X,Face),                 ;;; choose a face X
```

```
  Adj is adj_faces_of_face(X),   ;;; get a list of faces
                                 ;;; adjacent to face X
  member(Y,Adj),                 ;;; choose a face Y adjacent
                                 ;;; to face X
  insertor_type(X,Y),            ;;; check if X Y from an insertor
  give_ids([X,Y],I).             ;;; get the name for the compound
                                 ;;; feature

;;;INSERTOR-type-1
insertor_type(F1,F2) :-        ;;; F1,F2 are two instances
                               ;;; of face recordclasses
  T1 is face_type(F1),
  T1 = 1,                      ;;; F1 is cylindrical
  S is sign_of_face(F1),
  S = 1,                       ;;; F1 has a convex shape
  T2 is face_type(F2),
  T2 = 2,                      ;;; F2 is planar
  Edge1 is edges_of_face(F1),
  member(E,Edge1),             ;;; choose an edge E of F1
  Edge2 is edges_of_face(F2),
  member(E,Edge2),             ;;; E is also an edge of
                               ;;; face F2
  convex_edge(E,F1,F2),        ;;; E is a convex edge
  perp(F1,F2).                 ;;; F1, F2 are perpendicular
```

One can see that the process of finding a compound feature from a boundary representation of a body is to identify those faces that are pertinent to assembly planning. Several heuristics have been applied for increasing the efficiency during pattern matching, such as:

- Functional oriented search. Different subsections of the library are selected depending on what assembly goal is given. For example, *thread b1 b2* causes the system to look for feature patterns in a different subset than when *fit b1 b2* is given

- For a set of bodies, $B_1, B_2, ..., B_n$ with $m_1, m_2, ..., m_n$ as the number of faces of each body respectively, match faces of B_i against the compound feature library first where $m_i = min(\{m_1, ...m_n\})$.

- Let us define the degree of a compound feature F_{comp} as the number of primitive features of which is composed. After matching the above B_i a set of compound features $\{F_{comp\_i}\}$ of B_i is found, and there exists a $c \in \mathcal{N}$, a natural number, such that $c = max(\{D(F_{comp\_i})\})$. This number c sets an upper bound on the degree of the to-be-searched compound feature patterns for those bodies which are to be mated with B_i.

- Some primitive features such as cylindrical surface, surfaces with texture *threads*, or *gears* have higher priority to be matched first. Because such surfaces with given parameters are relatively rare, so the probability of being correctly matched is higher.

- When more than one compound feature is found for one body, match the compound feature which has the maximum degree first. If such a match leads to a consistent solution for the rest of the assembly, no further matching is pursued.

5.2 Compound feature computation : intersection of symmetry groups

Let us recall that a *compound feature* F_{comp} of body B is a set of primitive features F_i of B. Provided that F_is are all distinct (Section 4.3) the symmetry group of F_{comp} is

$$sym\_group(F_{comp}) = \bigcap_i sym\_group(F_i)$$

The algorithm developed in [13] has been applied to compute $sym\_group(F_{comp})$. In cases where $F_{comp} = \{F_1, F_2\}$, if F_1 and F_2 have the same symmetry group and the same dimensions, and are ***strongly congruent*** then the symmetry group of F_{comp} is the product of the above group intersection with a conjugated subgroup of $SO(3)$ derived from the permutation of the strongly congruent features (See proofs in section 4.3 Compound feature and its symmetry group). For example, one *multi-insertor* compound feature of $b1$ in Figure 1, which is composed of faces $1, 3$ and

4, has a rotational symmetry. This symmetry generates a cyclic group of order 2 about the axis which is parallel to the cylinders and goes through the center of line segment L (Figure 5).

In order to apply the theory to actual robotic reasoning, we make use of the boundary models provided within the POPLOG system by the linked-in PADL2 modeller. Each face F of a model is labeled with its symmetry group, each group being considered as the image $f^{-1}G_{canon}f$ of a canonical subgroup of $\mathcal{E}^+$ under an inner automorphism. A data-structure denoting the canonical subgroup G_{canon} is obtained by table lookup from the surface type of the face, using a POP-11 property procedure *gr_canon*. E.g. if F is a conical face *gr_canon*(F) is a data-structure denoting the group $SO(2)$ of all rotations about the Z-axis. The f for the inner-automorphism is the rigid transformation defining the location of the face in body coordinates as given by PADL2. It is possible to use the feature location because the way in which coordinate systems are embedded in features by PADL2 permits a coherent and consistent choice of canonical groups — largely this is because the Z-axis is chosen by PADL2 to be the axis of symmetry.

One important aspect of symmetry group intersection is to find the new coordinate system for the compound feature. Although the basic idea is to establish a coordinate system for the compound feature under which the symmetry group of the compound feature is in its canonical form, there can be multiple choices for the origin and orientations of certain axes. For example, when intersecting the symmetry groups of two parallel cylinders, if the two cylinders are 2-congruent then the choice of Z axis should be the line parallel to the center lines of the two cylinders and midway between them (Figure 5).

6 How to find mating features

6.1 Necessary mating conditions

When bodies are mated together in an assembly, certain relationships are established between their features. *Fitting*, for example, implies an *areal contact*, in other words, the two bodies in contact share at least one 2D surface. This means that the features over the area of contact have the same symmetry group. In this work we have restricted ourselves to consider just fitting relationships, which include *plane against plane* relationships since these involve areal contact. Thus one of the necessary conditions upon candidate mating features is that they must have the same symmetry group.

The concavity/convexity of each feature can be obtained from PADL2 or by a computation using data from the PADL2 boundary model. For two features to be mated they cannot both be concave (or both convex), except for planar features which are a special case.

Dimensional consistency of the candidate mating features is also required. There are two kinds of dimensions involved.

- The parameters of each PADL2 surface that is a component of one compound feature should be consistent with the parameters of the corresponding surface component of the other compound feature.

- The characteristic invariants used in calculating the intersection groups have intrinsic dimensions (e.g. the length of the common perpendicular between line invariants and the angle between them): these dimensions should be consistent between corresponding compound features.

Therefore the necessary conditions for a pair of features to be mating features are:

- They have the same symmetry group

- Their dimensions are consistent

6.2 Sufficient mating conditions

In addition to the geometry of the assembly, other physical constraints also pertain. Consider, for example, gravity: if the batteries are inserted into the tube of a flash light from below when it is in a vertical position with the open end downwards, they will simply fall out again due to gravity. In accordance with Newton's laws of motion, for bodies to be moved during actions, the robot must exert *forces* upon them. When it is necessary for bodies to remain in place, the planner must ensure that disturbing forces (usually gravity) are canceled out by reaction forces and friction. However, if the assembly is to be accomplished by a robot in space the physical constraints would be other than those arising from gravity on earth. Therefore the sufficient conditions for an assembly configuration are always relative to some pre-constraints. However, one basic constraint always pertains for assemblies of rigid bodies in Euclidean space, and that is the spatial occupancy constraint: no two bodies may occupy the same volume of space at the same time. The eligibility for two features from separate bodies to be mating features is affected by the rest of the (sub)assemblies as well. In order to check whether the spatial occupancy constraint is satisfied it is necessary to find relative positions of bodies in an assembly, which is the topic of the next subsection.

6.3 Relative positions of mating bodies and spatial interference checking

In this section we consider what we can infer about the *relative location* of two bodies that have two features in contact. We have noted that such a relation must correspond to a set of isometries which specify the relative location of the two bodies.

Now let B_1 and B_2 be two bodies, with primitive features F_1 and F_2 which have symmetry groups $sym(F_1), sym(F_2)$ and which are located in their respective body coordinate systems by isometries f_1 and f_2. Suppose the two features are in contact. If they are in contact over a finite area we say that F_1 *fits* F_2. If the contact is a line or point contact, then we say that F_1 *against* F_2. In either case, it is clear that if we move B_1 or B_2 by a member of the symmetry groups $sym(F_1)$ or $sym(F_2)$ respectively, the relationship between the features is preserved. We can express this most generally by a relationship between the isometries l_1, l_2 specifying the locations of bodies B_1, B_2 in the world coordinate system.

The *fits* relation is particularly constraining. If the primitive features are those algebraic sets which are used in PADL2, then areal contact implies that the surfaces are identical, so that the symmetry groups of contacting features are identical. For two algebraic sets to *fit*, one must be the complement of the other, and we have:

$$l_1^{-1} l_2 \in f_1 sym(F_1) f_2^{-1} \tag{1}$$

We can summarize this by saying that *if a primitive feature of one body fits a primitive feature of another body then the relative location of the two bodies is a two-sided coset of the common symmetry group of the features.*

In the case where $sym(F_1)$ is the identity group we have

$$l_1^{-1}l_2 = f_1 f_2^{-1} \tag{2}$$

The *intersect* predicate of PADL2 is then used for spatial occupancy checking whenever two candidate mating features have the *identity* group as their common symmetry group. For a complete spatial interference checking the number of operation is on the order of n^2 where n is the number of the bodies in the assembly.

Two bodies in an assembly are typically related to each other through multiple primitive features. If bodies B_1, B_2 are related by *fitting* two pairs of features, such as a peg in a blind hole i.e. f_{11} *fits* f_{21} and f_{12} *fits* f_{22} where f_{11}, f_{12} are features of B_1, f_{21}, f_{22} are features of B_2, we can use condition (1) to obtain the relative location $l_1^{-1}l_2$ of B_1 to B_2 as:

$$l_1^{-1}l_2 \in f_{11} sym(F_1) f_{21}^{-1} \bigcap f_{21} sym(F_2) f_{22}^{-1} \tag{3}$$

i.e. a member of the intersection of two-sided cosets. Since each two-sided coset can be rewritten as a one-sided coset

$$g_1 H g_2 = g_1 H g_1^{-1}(g_1 g_2) \tag{4}$$

where $g_1 H g_2$ is a two-sided coset of $H \subset G$, we can compute (3) as the intersection of two one-sided cosets.

$$l_1^{-1}l_2 \in sym(F_1')g_1 \bigcap sym(F_2')g_2 = (sym(F_1') \bigcap sym(F_2')g_2 g_1^{-1})g_1 \tag{5}$$

Further from the proposition 2 of [20] :

If H_1 and H_2 are subgroups of G and $g \in G$, then $H_1 \cap H_2 g$ is either null or is a coset of $H_1 \cap H_2$.

we have

$$l_1^{-1}l_2 \in (sym(F_1') \bigcap sym(F_2'))g \tag{6}$$

This says that the relative position of two bodies can be obtained by calculating the intersection of two conjugate groups of the symmetry groups and finding a particular element $g \in (sym(F_1') \bigcap sym(F_2')g_2 g_1^{-1})g_1$. This is solvable since we have developed an efficient group intersection algorithm for an adequate family of the subgroups of the Euclidean group, namely the TR subgroups [13]; and the method used in RAPT [4, 1] can be applied to find g.

7 Example: Three double pegs fit into another part with double holes

We now examine show how the planner finds mating features in the example of Figure 1. This example is drawn by and stored in PADL2.

The input goals are:

```
[goal 1 [fit b1 b4]]
[goal 2 [fit b2 b4]]
[goal 3 [fit b3 b4]]
```

Bodies $b1, b2$ and $b3$ have the same compound features $multi - insertor$:

```
b1 :
all_insertors--> [[1 3] [1 4] [3] [4]]
all_multi_insertors--> [[1 3 4] [1 4 3]]

comp1 : [1 3 4]

b2 :
all_insertors--> [[1 3] [1 4] [3] [4]]
all_multi_insertors--> [[1 3 4] [1 4 3]]
comp1 : [1 3 4]

b3 :
all_insertors--> [[1 2] [1 3] [2] [3]]
all_multi_insertors--> [[1 3 2]]
comp1 : [1 3 2]
```

Body *b4* has compound features *multi-container* which have the same degree as above:

```
b4 :
all_containers--> [[7 8] [7 9] [7 10] [7 11] [7 12] [7 13]]
all_multi_containers--> [[7 8 9] [7 8 10] [7 8 11] [7 8 12]
```

```
[7 8 13] [7 9 8] [7 9 10] [7 9 11] [7 9 12] [7 9 13] [7 10 8]
[7 10 9] [7 10 11] [7 10 12] [7 10 13] [7 11 8] [7 11 9]
[7 11 10] [7 11 12] [7 11 13] [7 12 8] [7 12 9] [7 12 10]
[7 12 11] [7 12 13]]

comp1 : [7 8 9]
comp2 : [7 8 10]
comp3 : [7 8 11]
comp4 : [7 8 12]
comp5 : [7 8 13]
comp6 : [7 9 10]
comp7 : [7 9 11]
comp8 : [7 9 12]
comp9 : [7 9 13]
comp10 : [7 10 11]
comp11 : [7 10 12]
comp12 : [7 10 13]
comp13 : [7 11 12]
comp14 : [7 11 13]
comp15 : [7 12 13]
```

Matching *b*1 and *b*4 gives:

```
in fit_matched !
symmetry group of comp1 of b1 : gr_identity
symmetry group of comp1 of b4 : gr_identity

Geometric data :
[4.0 [0 [1 <pl3 0.0 0.0 1.0 -4.0>]
        [1 <cyl3 1.0 <ln3 0.0 0.0 0.0 0.0 0.0 1.0>>]]
     [1 <cyl3 1.0 <ln3 4.0 0.0 0.0 0.0 0.0 1.0>>]]
[4.0 [0 [-1 <pl3 0.0 0.0 -1.0 11.0>]
        [-1 <cyl3 1.0 <ln3 5.0 2.0 11.0 0.0 0.0 1.0>>]]
     [-1 <cyl3 1.0 <ln3 9.0 2.0 11.0 0.0 0.0 1.0>>]]

b1 comp1 and b4 comp1 are matched !
```

Since the symmetry group of the mating compound mating feature is the identity group, the relative position of $b1$ and $b4$ connected by $b1$ compl and $b4$ compl is uniquely determined. If the position of $b4$ is assumed at the origin then the position of $b1$ is:

0.0	1.0	0.0	-5.0
1.0	0.0	0.0	-12.0
0.0	0.0	-1.0	15.0
0.0	0.0	0.0	1.0

Now the planner calls the PADL2 intersect predicate

```
result from PADL2 is
<false>
```

This says that there is no spatial interference detected. See Figure 7.

Similarly *compl* of $b1$ and *compl*3 of $b4$ are matched and spatial interference checking is also successful (Figure 8).

Matching $b2$ and $b4$ gives:

```
b2 compl and b4 compl are matched !
b2 compl and b4 compl3 are matched !
```

and matching $b3$ and $b4$ gives:

```
b3 compl and b4 compl2 are matched !
```

See Figure 9.

At this stage the planner has found all the compound features satisfying the spatial constraint locally. The last step before each assembly configuration is confirmed to be feasible is to check the spatial constraint globally (a constraint satisfaction network is applied [12]). Figure 10 shows that when the whole assembly is put together two interference volumes are found. Therefore the planner denies this configuration. After a modification to the shapes of the bodies $b1, b2$ and $b3$ by the user (designer) the planner confirms that a valid plan exists (11) and generates all the possible task specifications in terms of mating features:

```
** [Or  [And [fit [b1 comp1] [b4 comp13]]
             [fit [b2 comp1] [b4 comp1]]
             [fit [b3 comp1] [b4 comp12]]]
        [And [fit [b1 comp1] [b4 comp1]]
             [fit [b2 comp1] [b4 comp13]]
             [fit [b3 comp1] [b4 comp12]]]]
```

This result implies eight possible configurations of the given assembly since *comp*1 of *b*1 and *comp*1 of *b*2 both have the discrete rotational symmetries described at the end of Section 5.2. These reductions will be obvious when the output is re-written in terms of the primitive features composing each compound feature.

8 Discussion and Conclusion

The use of boundary models from a solid modeller as input to an assembly planner has the benefit of providing more coherent data than can be easily represented by special purpose descriptions. Solid modeller data can be obtained directly from a design database, and used further for geometric analysis and spatial reasoning. Using symmetry groups as feature descriptors allows us to treat features with computational uniformity. The common symmetry group of each mating feature pair can also be used for further analysis of the degrees of freedom within the assembly [26].

Searching through the whole feature library can be inefficient when there are many entries. One way to organize the library is to use compound feature degrees as an index. For body B with n surfaces, only those feature patterns P_i with $D(P_i) \leq n-1$ need to be matched.

One alternative way of combining compound features without using a feature library at all is to combine all the pairs or triples of each body. The total number of the compound features of degree 2 is $C_n^2 = n(n-1)/2$ which is on the order of n^2 where n is the number of primitive features on a body. This order is not very high but in practice the feature library method works faster in most of the cases. For $b4$ in Figure 1 there will be $(13 \times 12)/2 = 78$ such compound features, while there are only 15 compound features of $b4$ using our method.

When the highest degree compound feature pairs cannot be matched, matching proceeds in the direction of decreasing degree of compound feature patterns. In the worst case the primitive features will finally be matched against each other since these primitive features are the leaves of a compound feature tree. This can happen when mating bodies have some "exotic" feature patterns not in the library, when two bodies can only fit to each other through a pair of primitive features, or when there is no match at all. It would be desirable that the missing patterns are recognized and added to the library after one trial. The current system does not have this learning ability. A desirable extension of this system would be learning new, relevant, compound feature patterns to add to the library.

This work has shown the advantage of combining common sense knowledge with a rigorous mathematical treatment of object representation in assembly planning. It brings us the merits of flexibility in problem solving along with computational tractability. However, in order to handle a larger set of assembly tasks more work needs to be done. For example, the current version of the planner only treats those relations which have an areal contact. Group products have to be handled if line and point contacts are involved. Some of the special cases of such relationships have been treated in RAPT [4, 1].

9 Appendix

Definition 11 *A polynomial* $p \in F[x]$ *is said to be* **irreducible over F** *or* **irreducible in** $F[x]$ *or* **prime in** $F[x]$ *if* p *has positive degree and* $p = bc$ *with* $b, c \in F[x]$ *implies that either* b *or* c *is a constant polynomial.*

These are some topology definitions and theorems from [5, 16]. They are listed here to help readers to understand some relevant proofs in this chapter.

Definition 12 *A* **topology** *for a set* X *is a family* T *of subsets of* X *satisfying the following three properties:*

- The set X and the empty set $\emptyset$ are in T.
- The union of any family of members of T is in T.
- The intersection of any finite family of members of T is in T.

Definition 13 *The members of* T *are called* **open sets**.

Definition 14 *A* **neighborhood** *of a point $x \in X$ is an open set containing x.*

Definition 15 *A point x is a* **limit point** *of a subset A of X means that every neighborhood of x contains a point of A distinct from x. A* **closure** *of a set A is the set $\bar{A}$, the union of A with its set of limit points. The* **boundary** *of A is the intersect of $\bar{A}$ with $\overline{X \setminus A}$.*

Definition 16 *A* **path**, *is a space $[X, \mathcal{O}]$ is a mapping $p : [a, b] \to X$, where $[a, b]$ is a closed interval in $\Re$. If $p(a) = P$ and $p(b) = Q$, then p is a* **path from** P **to** Q.

Definition 17 *Two sets H, K are* **separated** *if*

$$\bar{H} \cap K = H \cap \bar{K} = \emptyset.$$

Theorem 1 *A set $M \subset X$ is* **connected** *if and only if M is not the union of two nonempty separated sets.*

Theorem 2 *For sets, connectivity is preserved by surjective mappings.*

Theorem 3 *If H and K are separated, then every connected subset M of $H \cup K$ lies either in H or in K.*

10 Acknowledgement

We would like to thank Gordon Dakin for implementing the POPLOG to PADL2 interface, and Richard Weiss for some very helpful discussions. Preparation of this paper was supported in part by NSF grant number IRI-8709949, in part by ONR grant number N00014-84-K-0564 and grant number N00014-86-K-0764, and in part by Philips Laboratories, North American Philips Corporation at Briarcliff Manor, NY.

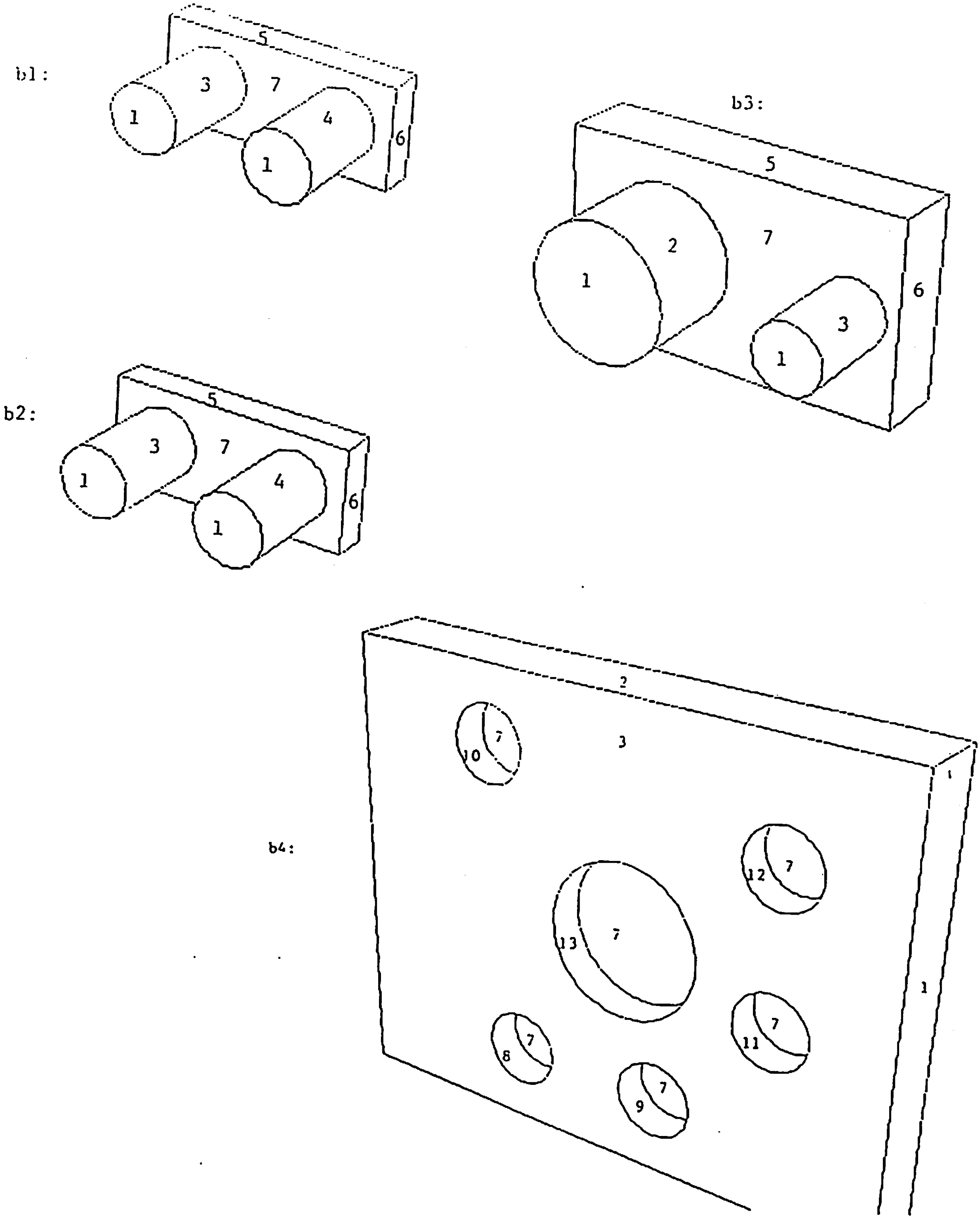

Figure 1: A simple assembly example

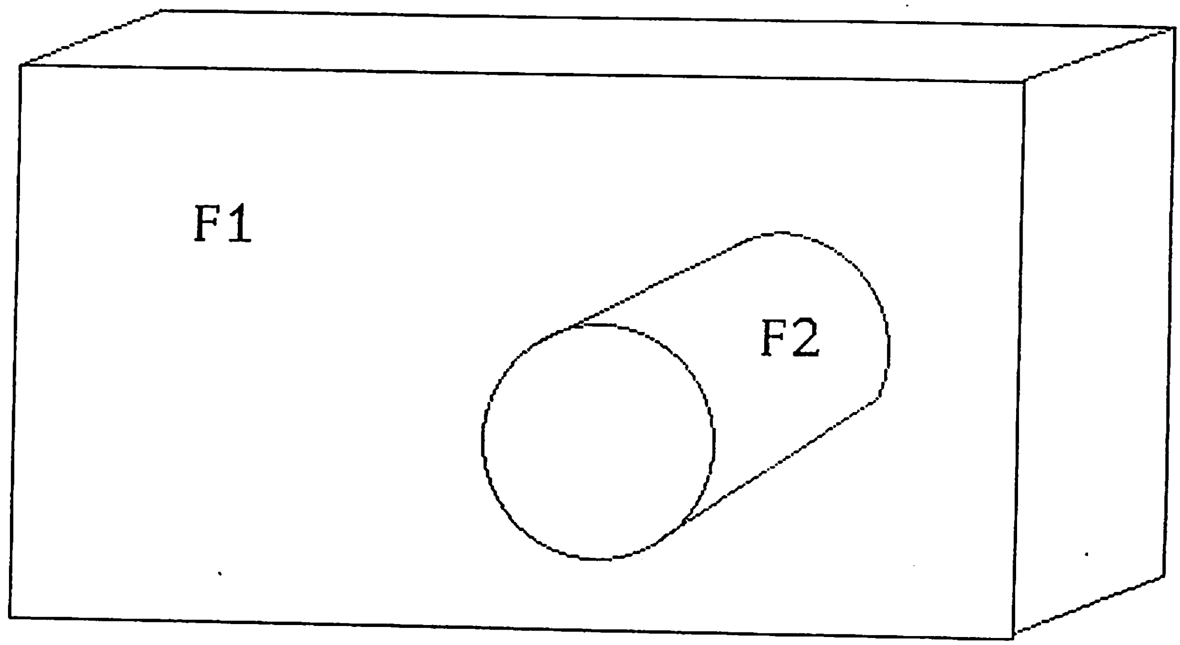

Figure 2: A pair of distinct features F_1, F_2

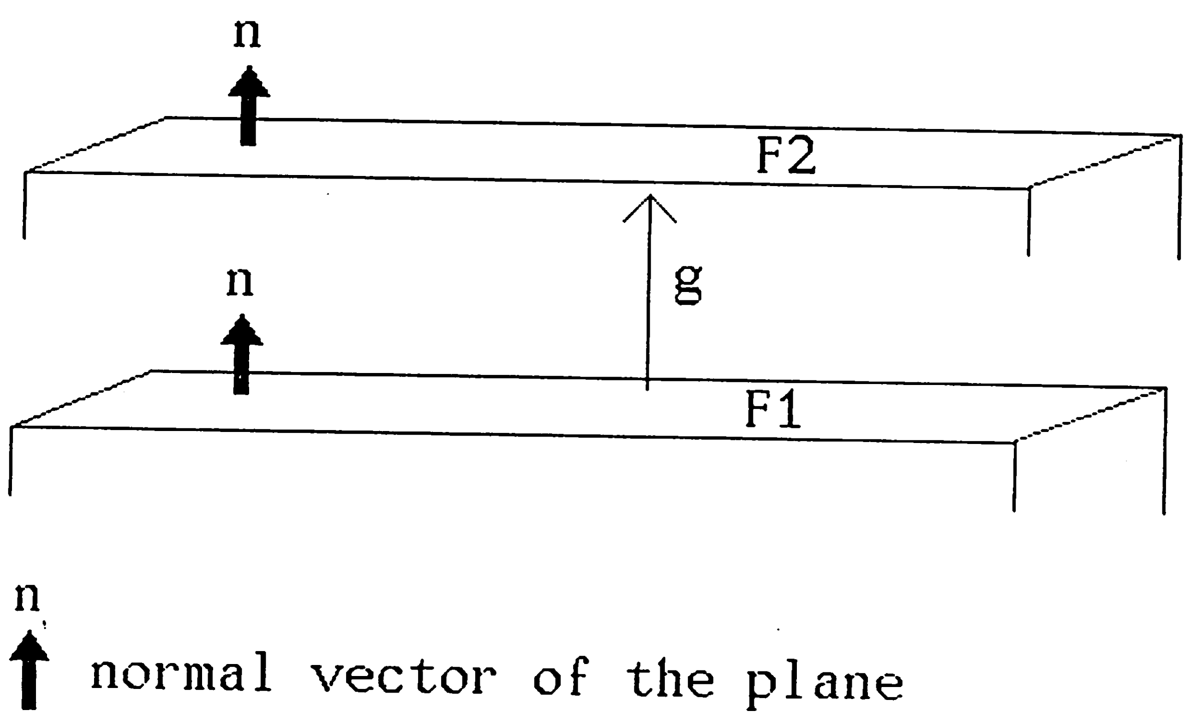

Figure 3: Two directed plane features F_1, F_2 which are weakly congruent of each other

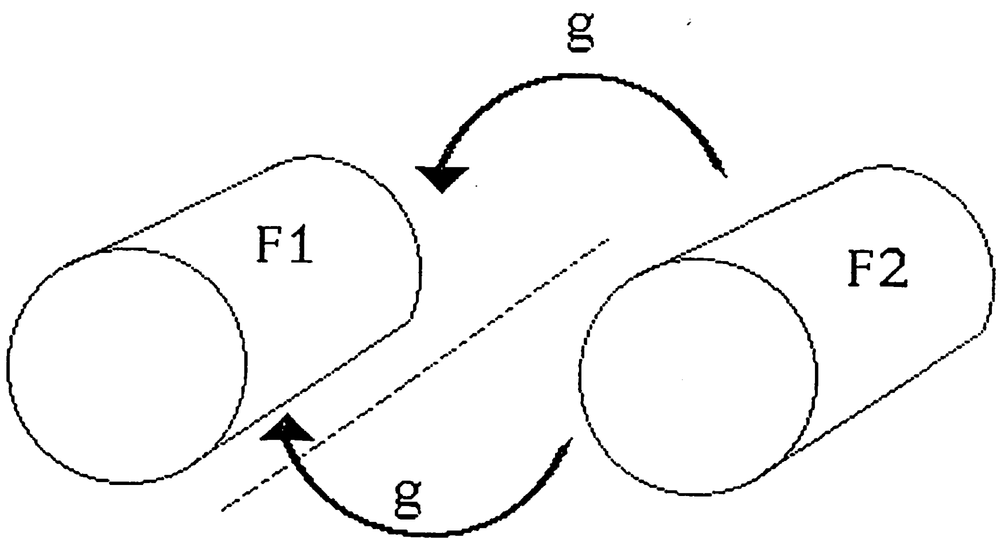

Figure 4: Two cylindrical features F_1, F_2 which are strongly congruent of each other

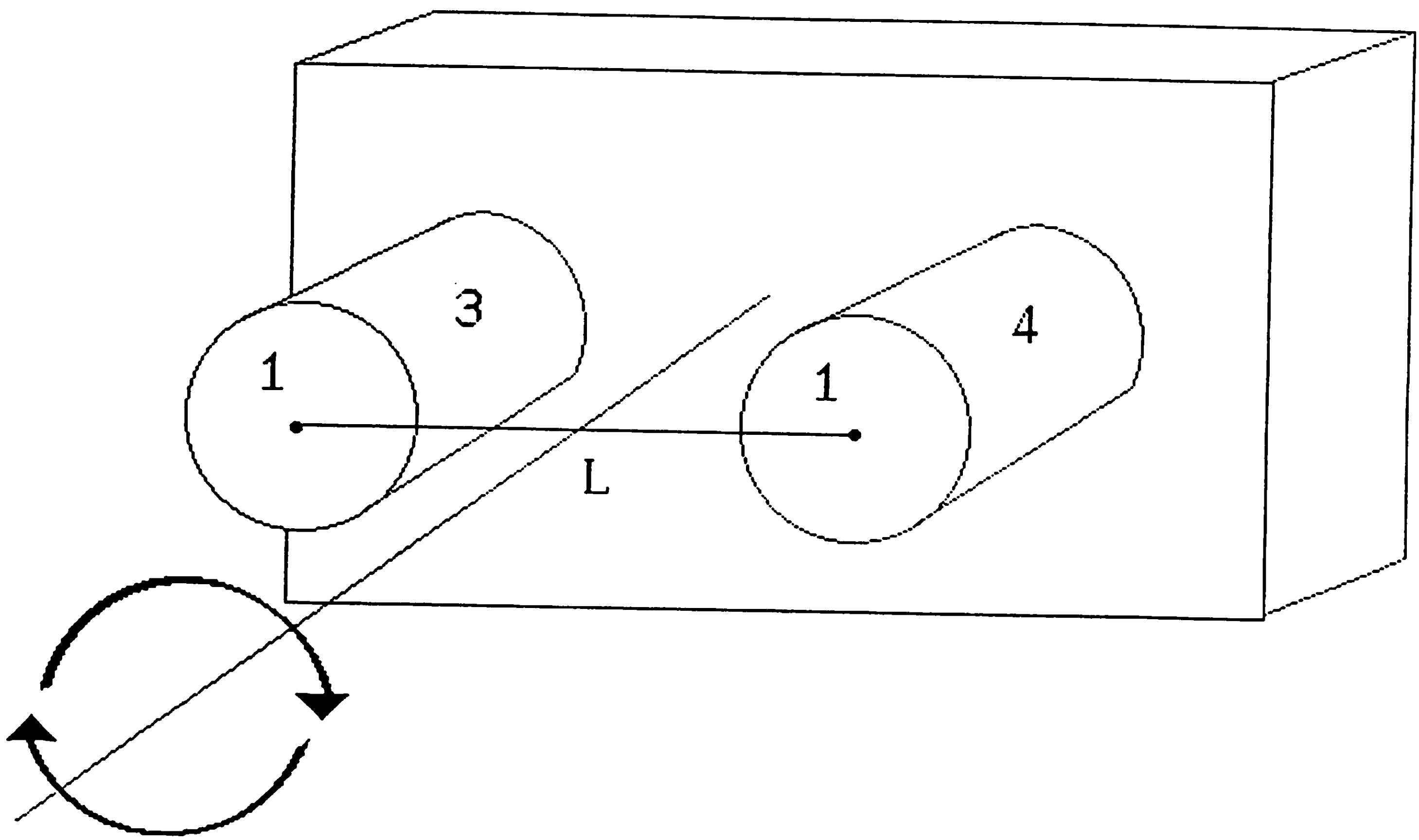

Figure 5: Extra symmetry in a compound feature

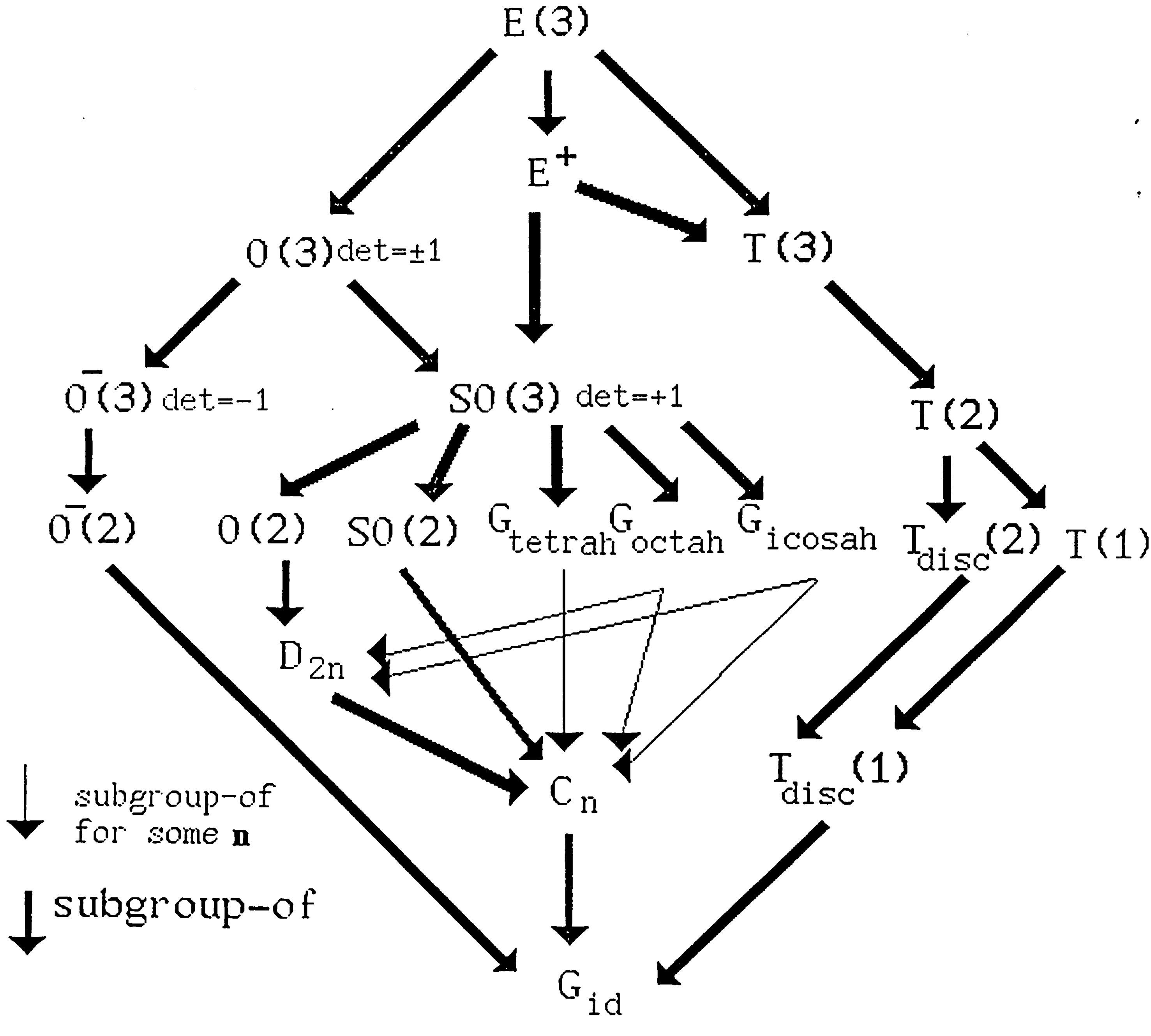

Figure 6: Relationships among some important subgroups of $\mathcal{E}^+$

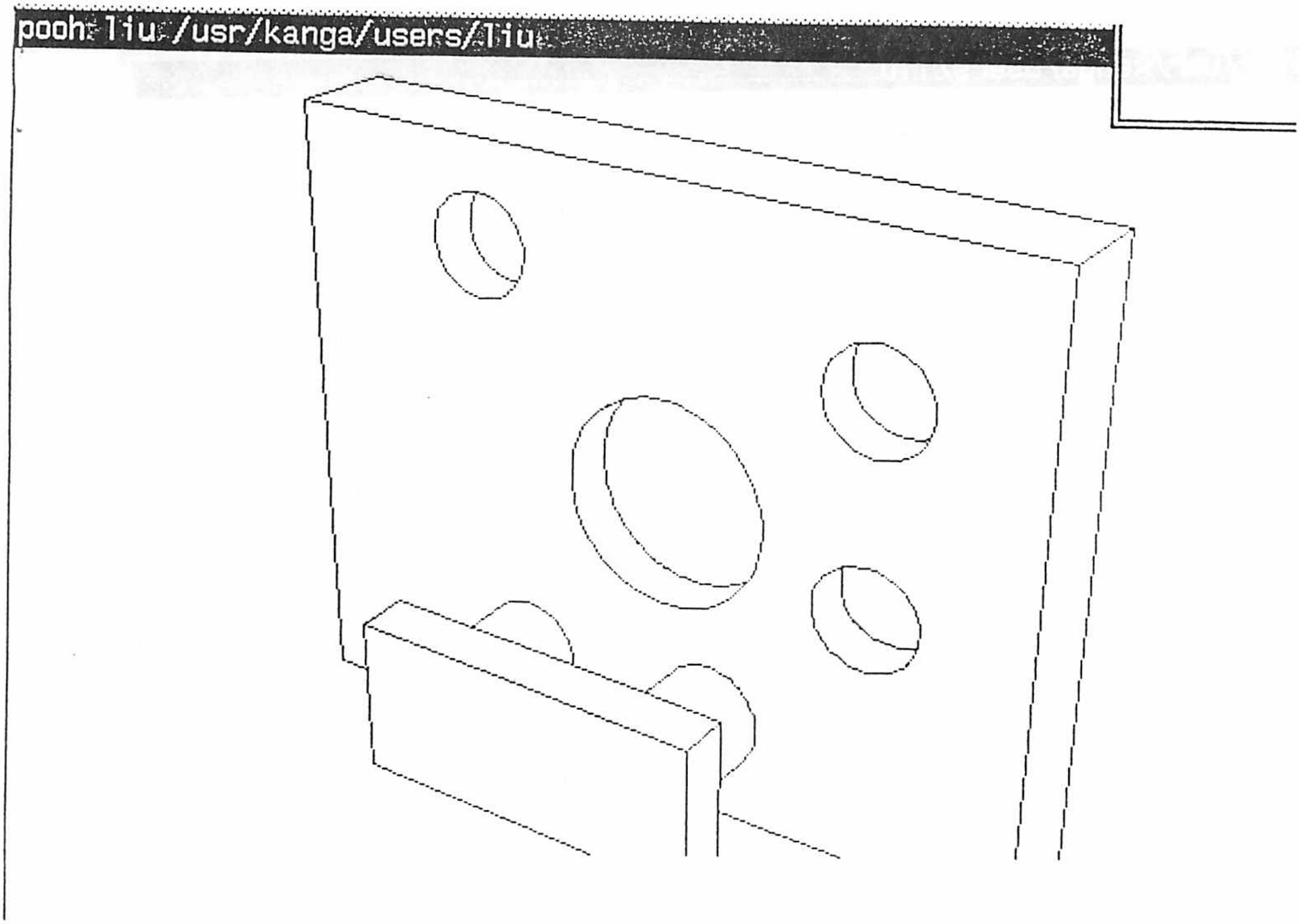

Figure 7: Body b1 and body b4 in one assembled position

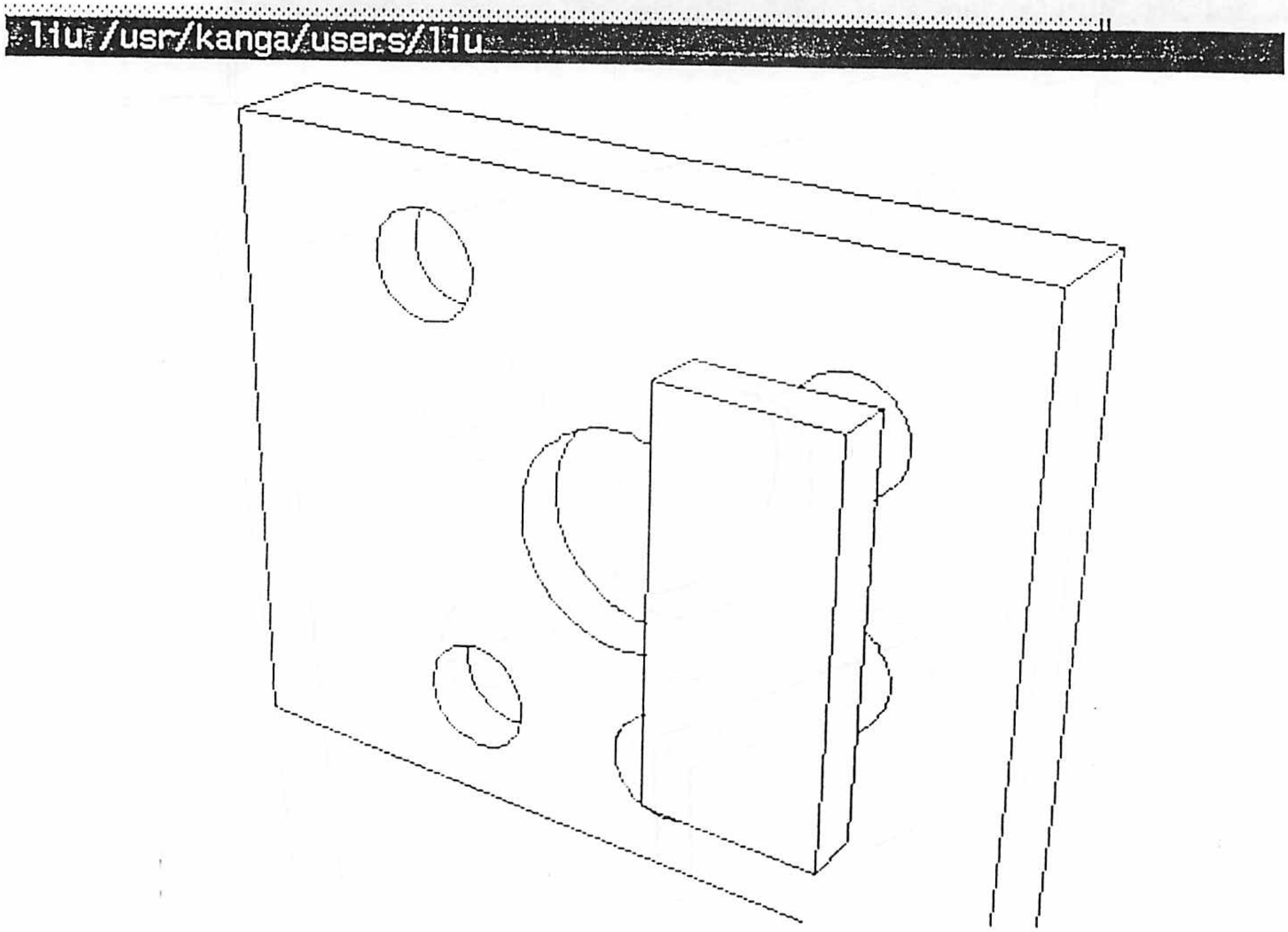

Figure 8: Body b1 and body b4 in another assembled position

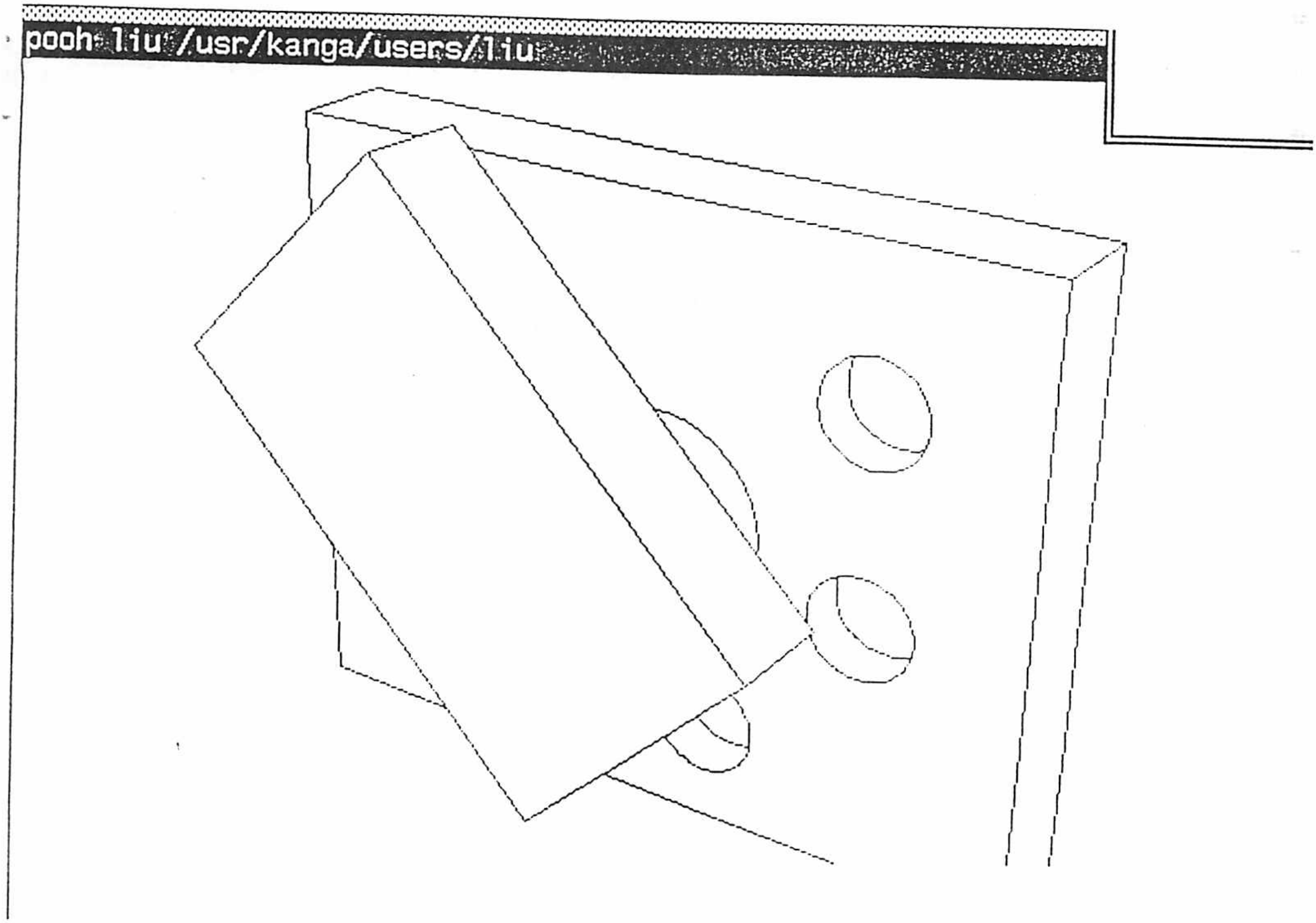

Figure 9: Body b3 and body b4 in an assembled position

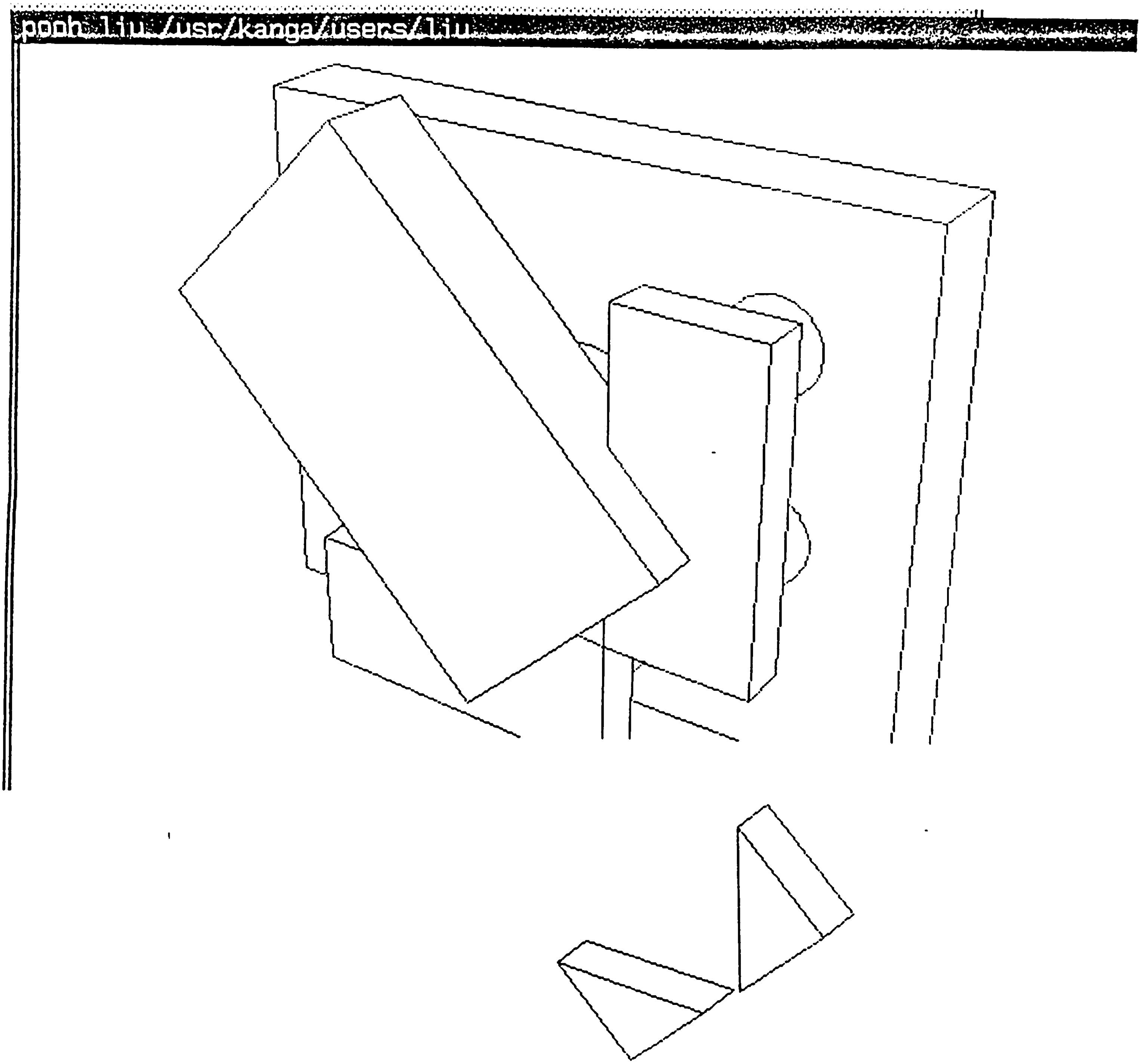

Figure 10: All four bodies in their assembled position — intersection detected

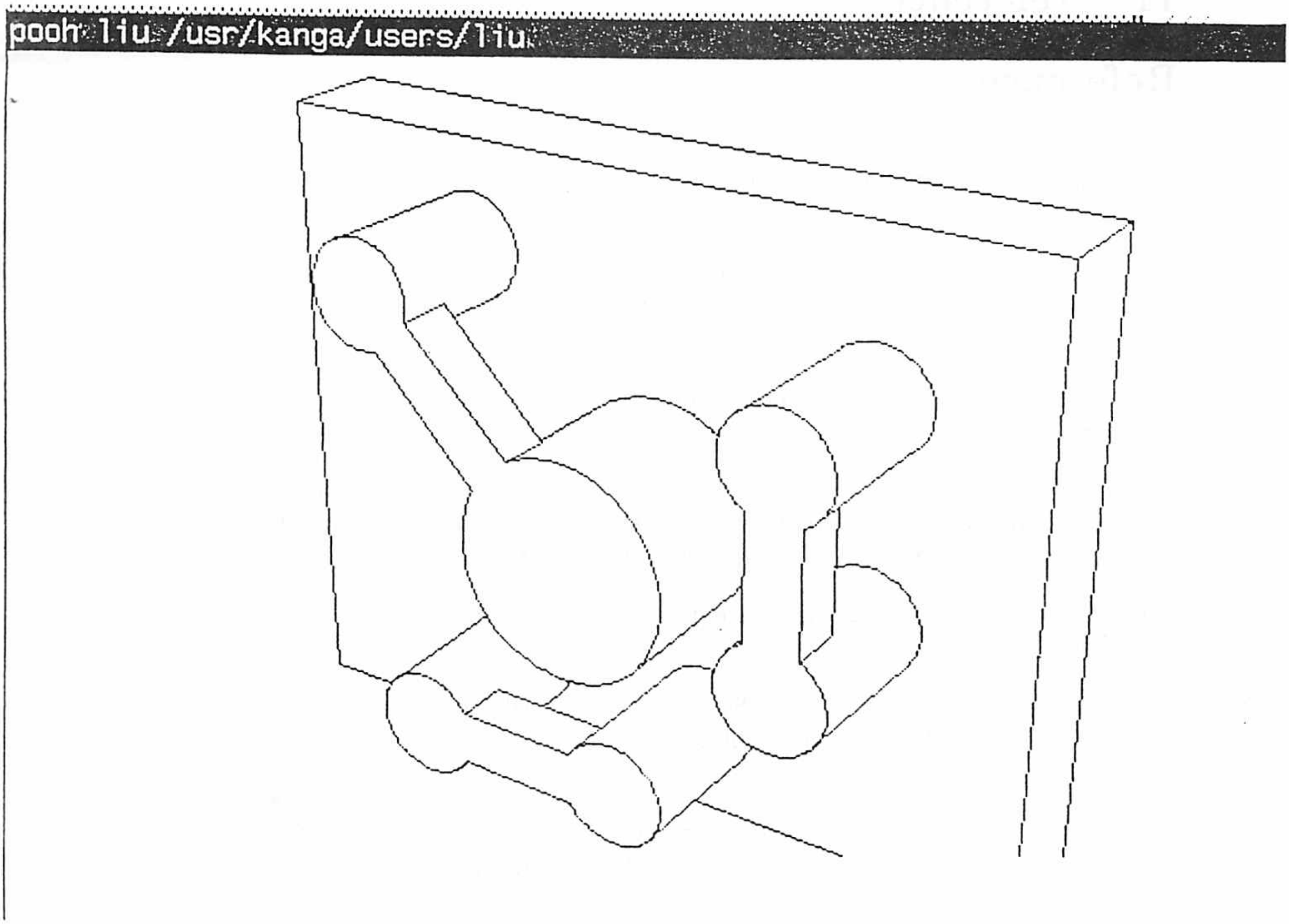

Figure 11: Revised four bodies in their assembled position

11 References

References

[1] Ambler, A.P. and Popplestone, R.J., (1975), "Inferring the positions of Bodies from specified Spatial Relationships", *Artificial Intelligence*, 6(1975), pp157-174.

[2] Brown, C.M., "PADL2: A Technical summary." *IEEE Computer Graphics Appl.*, vol. 2, no. 2, pp. 69-84, March, 1982.

[3] Carney, P.C. and Brown, D.C. (1988), "A Continued Investigation into Qualitative Reasoning about Shape and Fit", submitted to *AI EDAM* journal, Nov. 1988

[4] Corner D.F.; Ambler A.P. and Popplestone R.J., 1983, Reasoning about the Spatial Relationships Derived from a RAPT Program for Describing Assembly by Robot, Proc 8th. IJCAI, Karlsruhe,DBR., pp 842-844.

[5] Croom, F.H. (1978) "Basic Concepts of Algebraic Topology", Springer-Verlag, New York.

[6] Dixon, J.R. and Dym, C.L. [1986], "Artificial Intelligence and Geometric Reasoning in Manufacturing Technology",*Appl. Mech. Rev*, Vol. 39, No. 10.

[7] Green, D.S. and Brown, D.C. (1987), "Qualitative Reasoning during design about Shape and Fit: a preliminary report", *Expert Systems in Computer-Aided design*, J.S. Gero, north-Holland, Amsterdam, pp. 93-112.

[8] Hardy S., 1984, A New Software Environment for List-Processing and Logic Programming, in Artificial Intelligence, Tools, Techniques and Applications, eds. O'Shea,T. and Eisenstadt,M., Harper and Row, N.Y..

[9] Henderson, M.R. (1984) "Extraction of Feature Information from Three Dimensional CAD Data", Ph.D thesis, Department of Mechanical Engineering, Purdue University.

[10] Hutchinson, S.A. and Kak, A.C. (1990) "SPAR: A Planner That Satisfies Operational and Geometric Goals in Uncertain Environments", *Artificial Intelligence Magazine*, Spring, 1990.

[11] Liu, Y. and Popplestone, R.J. (1989), "Assembly Planning from Solid Models", the proceedings of *IEEE International Conference on Robotics and Automation*, Scottsdale, Arizona, May 14-19, 1989.

[12] Liu, Y. and Popplestone, R.J. (1990), "Symmetry Constraint Inference in Assembly Planning", *AAAI-90*, the proceeding of the Eighth National Conference on Artificial Intelligence, to appear, July 29 august 3, Boston, Mass.

[13] Liu, Y. (1990) "A Justification for the Characteristic Invariant Representation of TR Subgroups of the Proper Euclidean Group", to be presented at the fifth SIAM Conference on Discrete Mathematics, June 11 -14, Atlanta, Georgia.

[14] Lozano-Pérez, T. (1982), "Task Planning", *Robot Motion: Planning and Control*, Ed. Michael Brady, et al, The MIT Press Series in Artificial Intelligence

[15] Lozano-Pérez, T., et al. *Handey: A Robot System that Recognizes, Plans, and Manipulates*, the proceedings of IEEE International Conference on Robotics and Automation, Raleigh, North Carolina, March 31-April 3, 1987.

[16] Moise, E.E. (1977) "Geometric Topology in Dimensions 2 and 3", Springer-Verlag, New York.

[17] Miller,W. Jr. (1972), "Symmetry Groups and Their Applications", *Academic Press* New York and London, 1972.

[18] Nevins, J.L. and Whitney, D.E., "Assembly Research", *Automatica*, Vol 16, number 6.

[19] Oussama, K, Craig, J.J and Lozano-Pérez, T. Eds. (1989) "The Robotics Review", the MIT Press, Cambridge, Massachusetts, London, England.

[20] Popplestone, R.J. (1984), "Group theory and Robotics", *Robotics Research, The First Int. Symp.*, M. Brady and R. Paul, Eds. Cambridge, Ma, MIT Press, 1984.

[21] Popplestone, R.J. 1988 "The Edinburgh Designer System as a Framework for Robotics: The Design of Behavior", AI EDAM, Vol 1 No 1.

[22] Popplestone R.J. , Weiss R., and Liu Y. *Using Characteristic Invariants to Infer New Spatial relationships from Old*, The proceedings of the 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988.

[23] Popplestone, R.J., Liu, Y. and Weiss, R. (1990) "A Group Theoretic Approach to Assembly Planning", *Artificial Intelligence Magazine*, Spring, 1990.

[24] Requicha, A.A.G. and Voelcker, *Solid Modeling: Current Status and Research Directions*, IEEE Computer Graphics Applications, vol. 3, no. 7, pp25-37, October, 1983.

[25] Sanderson, A.C. and Homem-de-Mello, L.S., *Task Planning and Control Synthesis for Flexible Assembly Systems*, NATO International Advanced Research Workshop on Machine Intelligence and Knowledge Engineering for Robotics Applications, Maratea, Italy, May 12-16, 1986.

[26] Thomas, F. and Torras, C., (1988), "A Group-Theorectic Approach to the Computation of Symbolic Part Relations", *IEEE Journal of Robotics and Automation*, vol.4, number 6, December, 1988.

[27] Woodbury,R.F. and Oppenheim,I.J. (1988) "An Approach to Geometric Reasoning in Robotics", IEEE Transactions on Aerospace and Electronic Systems.