SYMMETRY GROUPS IN ROBOTIC ASSEMBLY PLANNING

Yanxi Liu

Computer and Information Science Department
University of Massachusetts

COINS Technical Report 90-83

Computer and Information Science

University of Massachusetts at Amherst

SYMMETRY GROUPS IN ROBOTIC ASSEMBLY PLANNING

Yanxi Liu

Computer and Information Science Department
University of Massachusetts

COINS Technical Report 90-83

SYMMETRY GROUPS IN ROBOTIC ASSEMBLY PLANNING

A Dissertation Presented

by

YANXI LIU

Submitted to the Graduate School of the
University of Massachusetts in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

SEPTEMBER 1990

Department of Computer and Information Science

This research was supported in part by the National Science Foundatio grant number IRI-8709949, in part by ONR grant numbers N00014-84-K-0 N00014-86-K-0764, in part by University Research Initiative Grant N0001 0764, and in part by Philips Laboratories, North American Philips Corpor Briarcliff Manor, NY.

SYMMETRY GROUPS IN ROBOTIC ASSEMBLY PLANNING

A Dissertation Presented

by

YANXI LIU

Approved as to style and content by:

Robin J. Popplestone, Chair

W. Bruce Croft, Member

John R. Dixon, Member

Roderic A. Grupen, Member

Robert N. Moll, Member

Marjorie Senechal, Member

W. Richards Adrion, Department Chair
Computer and Information Science

ACKNOWLEDGEMENT

At this point, I understand why people often say *a Ph.D. thesis is not a single person's work.* During all these years, I have been taught, advised and helped by lots of people around me. I would like to thank them all. There are a few I would like to single out in particular:

First of all, I want to thank my advisor Professor Robin J. Popplestone. His insightful guidance is an inseparable part of this thesis. He challenged me to learn more mathematics, for that I feel very grateful. He also challenged me to read *Winnie the Pooh*; I thank him for that too.

Professor John Dixon is the person who introduced to me the usage of symmetries in mechanical design, which started my fancy with symmetries.

I am very fortunate to get to know Professor Morjorie Senechal who gave me valuable encouragement and help on Crystallographic groups at the time I needed them the most.

Rod Grupen's enthusiastic attitude towards work and life has helped me several times to look forward

Thanks to all the members of my thesis committee for having faith in me and interest in my work. Their questions and suggestions on an earlier version of this thesis were very helpful.

Thanks to Rich Weiss and John Oliensis for being always ready to answer any questions I came up with concerning topology.

I would like to thank Professor M. Arbib for starting my interest in robotic assembly planning.

Thanks to Damian Lyons for being my mentor when I worked as a summer intern at Philips Labs. That experience broadened my view of the field.

I thank all my colleagues, especially, Gordon Dakin for his solid work.

My work has also been improved through discussions with the following people: George Avrunin, Mark Snyder, Seth Malitze, Susan Landau, King Shaw, Glenn Kramer, Jan Wolter, Avi Kak, Xu Yangsheng, Federico Thomas, Harpreet, Ruki, A. Zahnd, ...

Thanks to whoever for the fact that mathematics is nobody's private property. I thank everybody who has spread the beauty of math.

I wish my father could read this and be pleased that I remembered his last words: "study well ...". My dear mother and granny have given me their blessings for whatever I wish to pursue though it may take me far away from home. It is the strength I gained from them that gave me the will and the courage to go through all the good and bad times to get here. This dissertation is dedicated to them with affection.

Thanks to my husband's family for their understanding and support, and their stuffing monthly birthday-cake parties.

I thank Mr. and Mrs. Leo Collins for all the pleasant moments they brought me.

I want to thank all my friends and relations for the laughters and friendship we shared. Thanks to all my tennis partners for the sheer pleasure.

I'd like to thank Glenn, Pat, Rose, Renee, Sue, Darlene, Janet, Laurie and Sharon for their fine secretary/accounting work which saved me from despair at one time or another.

Last but never the least, without the love and support from my husband, my dearest friend, an inventive cook and an excellent reader/writer, Bob Collins, I could not have done what is happening here.

ABSTRACT

Symmetry Groups in Robotic Assembly Planning

September 1990

YANXI LIU

B.S., Beijing Normal University

M.S., University of Massachusetts at Amherst

Ph.D., University of Massachusetts at Amherst

Directed by: Professor Robin J. Popplestone

Human beings have long appreciated symmetries of objects. However, little of this appreciation has been addressed formally and computationally in today's automatic systems that deal with engineered bodies. In this work group theory, being the standard mathematical tool for describing symmetry, is applied to assembly planning for robotics in a computationally tractable way.

Using group theory to characterize the assembly of bodies and features, especially the symmetries relevant to contact between bodies, reveals the necessity of a group intersection algorithm for intersecting subgroups of the proper Euclidean group. The central results of this thesis are to establish this necessity mathematically and provide a compact representation for the subgroups of the Euclidean group that lends itself to the implementation of an efficient group intersection algorithm.

I define a geometric representation in terms of *characteristic invariants* for an important family of subgroups of the proper Euclidean group. Each member of this family is called a TR group since it is a semidirect product of a translation group T and a rotation group R. Some examples of TR groups are the translation subgroups T^1, T^2, and T^3; the rotation subgroups $SO(2), SO(3), O(2)$, the cyclic groups, the dihedral groups, and the platonic groups; as well as groups containing both rotations and translations such as the symmetry groups of the plane and infinite cylinder, and certain Crystallographic groups. I prove that there is a one-to-one correspondence

between TR groups and their characteristic invariants. I also prove that the intersection of TR groups is closed and can be calculated from their characteristic invariants. An efficient intersection algorithm using characteristic invariants has been implemented.

A practical issue addressed in this dissertation is the linkage between the mechanical design and the robotic task-level planning. The formal treatment of TR symmetry groups has been embedded into the implementation of an assembly planning system $\mathcal{KA}3$. $\mathcal{KA}3$ takes as part of its input the geometric boundary models of assembly components provided by a geometric solid modeller PADL2. Being able to represent and reason about symmetries associated with bodies and features, $\mathcal{KA}3$ finds a set of detailed robotic assembly task specifications in three steps:

Step one: $\mathcal{KA}3$ finds mating features from the boundary models of assembly components using a salient feature library and the symmetry group intersection algorithm developed in this work. This permits much simpler task specifications to be used. Step two: $\mathcal{KA}3$ applies techniques used in constraint satisfaction problems (CSP) to satisfy the kinematic and spatial constraints for each candidate assembly configuration. Step three: $\mathcal{KA}3$ generates a partially ordered sequence of contact states for assembly components through an analysis of disassembly via translational motion.

The interaction between algebra and geometry within a group theoretic framework and the interaction between CSP techniques and heuristic search strategies provides us with a unified computational treatment of reasoning about how parts with multiple contacting features fit together.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS v

ABSTRACT vii

TABLE OF CONTENTS ix

LIST OF TABLES xiii

LIST OF FIGURES xiv

CHAPTERS

1. **INTRODUCTION** 1
   - 1.1 Statement of the Problem 5
   - 1.2 $\mathcal{KA}3$ is an Interdisciplinary System 8
   - 1.3 Organization of the Dissertation 10
2. **RELATED WORK** 12
   - 2.1 Planning in Artificial Intelligence and Planning in Robotics 12
     - 2.1.1 Domain Independent Planning in AI 13
     - 2.1.2 Planning in Robotics 13
     - 2.1.3 Assembly Planning for Robotics 15
   - 2.2 Mechanical Design and Automatic Assembly 15
     - 2.2.1 Design for Assembly 15
     - 2.2.2 Automatic Assembly 16
   - 2.3 Geometric Reasoning 16
     - 2.3.1 From Spatial Relationships to Relative Positions 16
     - 2.3.2 Feature Inferencing 17

2.3.3 Degree of Freedom Analysis 18
2.4 General Motion Planning versus Contact Motion Analysis 18
2.5 Applying Group theory in Robotics 20
2.5.1 Group Theory and Mechanisms 20
2.5.2 Group Theory and Robotics 20
2.5.3 Computational Treatment of Groups 21

3. FEATURES, SYMMETRIES and SYMMETRY GROUPS 23
3.1 Abstract Groups and the Proper Euclidean Group 23
3.2 Primitive Features and Their Symmetry Groups 25
3.3 Compound Features and Their Symmetry Groups 30
3.4 Symmetry Groups at Areas of Contact 33
3.5 Discussions on Oriented Features 35
3.6 Summary 38

4. TR SUBGROUPS OF THE PROPER EUCLIDEAN GROUP 40
4.1 Preliminaries 43
4.1.1 Abstract Groups 43
4.1.2 Group Actions 43
4.1.3 The Euclidean Group and its Subgroups 46
4.2 Translational Characteristic Invariants for Translation Subgroups 58
4.3 Rotational Characteristic Invariants for Rotation Subgroups 59
4.4 **TR** Subgroups of $\mathcal{E}^+$ 67
4.4.1 Translational Characteristic Invariant for **TR** subgroups 68
4.4.2 Crystallographic Groups 74
4.4.3 Rotational Characteristic Invariants for **TR** subgroups 79
4.4.4 **TR** Group Intersection 82
4.5 An Algorithm Outline for TR Group Intersection 85

4.6 Summary . . . 87

5. AN ASSEMBLY PLANNING SYSTEM $\mathcal{KA}3$ . . . 88

5.1 Assembly Feature-mating Inference from Solid Models Using Symmetry Groups . . . 90

5.1.1 How to Describe Shapes to PADL2 . . . 90

5.1.2 How PADL2 Represents Solids . . . 91

5.1.3 Compound Feature Recognition and Computation . . . 93

5.1.4 How to Find Mating Features . . . 98

5.2 Assembly Planning via Kinematic and Spatial Constraint Satisfaction 99

5.2.1 Symmetry Constraints and Constraint Satisfaction Process . . . 100

5.3 Disassembly-motion Analysis . . . 103

5.3.1 Degree of Freedom Analysis . . . 104

5.3.2 Spatial Interference Checking . . . 105

5.3.3 Disassembly Order Analysis . . . 106

5.4 Experiments on $\mathcal{KA}3$ . . . 108

5.4.1 An Assembly with Double-pegs and Multiple Holes . . . 108

5.4.2 An Assembly with All Revolute Joints . . . 112

5.4.3 An Assembly with Both Discrete and Continuous Joints . . . 113

5.4.4 A Gearbox . . . 114

5.5 Discussion . . . 116

6. CONCLUSION and FUTURE WORK . . . 131

6.1 Contributions . . . 131

6.2 Future Research . . . 132

6.3 Remarks . . . 133

APPENDICES

A. ABSTRACT ALGEBRA . . . 135

B. POINT-SET TOPOLOGY 136

C. FINITE ROTATION GROUPS 138

D. INTERSECTION OF DISCRETE TR GROUPS 139

E. IMPLEMENTATION OF SHAPES USING PADL2 141

F. DEFINITION FOR EDGES AND GENERAL FEATURE TYPES 143

G. WHY IS THE PLANNING SYSTEM CALLED $\mathcal{KA}3$? 144

BIBLIOGRAPHY 146

LIST OF TABLES

2.1 Continuous Groups and their Degrees of Freedom . . . 19

2.2 Characteristic Invariants of Important Subgroups of $\mathcal{E}^+$ . . . 22

3.1 Some important subgroups of $\mathcal{E}^+$ . . . 28

3.2 Correspondence between shapes and their symmetry groups . . . 29

3.3 Continuous Group and Degree of Freedom . . . 36

4.1 The **TR** subgroups in one and two dimensional Crystallographic Groups 75

4.2 The **TR** subgroups in three dimensional Crystallographic Groups . . 75

5.1 Some typical constraints given by users . . . 90

5.2 Possible Motions versus Restricted Motions . . . 106

C.1 Types of finite rotation groups . . . 138

E.1 Infinite Solids and their descriptions . . . 142

LIST OF FIGURES

1.1 Platonic solids 2

1.2 Lower Pairs 3

1.3 A block with a cylindrical hole 7

3.1 A pair of **distinct** features F_1, F_2 26

3.2 Two directed plane features F_1, F_2 which are **weakly congruent** to each other 27

3.3 Two cylindrical features F_1, F_2 which are **strongly congruent** to each other 27

3.4 Relationships among some important subgroups of $\mathcal{E}^+$ 29

3.5 Distinct features and relations among their open sets 31

4.1 C_4 does not act on X transitively — there are two disjoint orbits; C_4 acts on Y transitively — there is only a single orbit 45

4.2 Identifying $\mathbf{T}^3$ with $\Re^3$ 49

4.3 This shows the effect of trt^{-1} acting on an arbitrary point x in order to prove that trt^{-1} is a rotation about axis $t(\bar{r})$ with angle of rotation θ_r 53

4.4 Minimum translation t_{min} 55

4.5 When a rotation group has only two poles p_1 and p_2, they have to belong to the same rotation axis; otherwise more than two poles would appear 56

4.6 Examples for the fixed point set $\mathcal{F}_R$ and pole-set $\mathcal{P}_R$ of some rotation subgroups R 61

4.7 When $\mathcal{F} \neq \emptyset$, $R = t_{\mathcal{F}} R_c t_{\mathcal{F}}^{-1}$, $R_1 = t_{\mathcal{F}} R_{c_1} t_{\mathcal{F}}^{-1}$ and $R_2 = t_{\mathcal{F}} R_{c_2} t_{\mathcal{F}}^{-1}$ 64

4.8 $R = R_1 \cap R_2$ is a line group. All the rotations in R fix both points $\mathcal{F}_{R_1}$ and $\mathcal{F}_{R_2}$. Thus they fix the line $\mathcal{L}$. So $\mathcal{F}_R = \mathcal{L}$ 66

4.9 When R_1, R_2 are both point groups and $\bar{g} \perp \mathcal{T}_G$ 72

4.10 The seventeen two-dimensional Crystallographic groups 73

4.11 The Crystallographic group $p4$ characterised by $\mathcal{F}_{\tilde{R}}$ and $\mathcal{P}_{\tilde{R}}$ 76

4.12 Rotation symmetries for an infinite cylinder: S_0 can be moved along $T(\mathcal{F}_{R_{cyl}})$ 80

4.13 $G_1 = TR_1 = TR_2$. Since $\mathcal{F}_{R_1}$ and $\mathcal{F}_{R_2}$ each denotes a set of 4 points (lines), from $T(\mathcal{F}_{R_1}) = T(\mathcal{F}_{R_2})$ it is not necessarily true that there exists $t \in T$ such that $\mathcal{F}_{R_1} = t(\mathcal{F}_{R_2})$. Here t_1, t_2 are generators of T. 81

4.14 When R_1, R_2 are both point groups and $\bar{g} \perp \mathcal{T}_G$ 83

4.15 When T_1, T_2 are both one dimensional discrete groups 84

4.16 The outline of the **TR** group intersection algorithm 86

5.1 An overview of $\mathcal{KA}3$ system 89

5.2 Extra symmetry in a compound feature 97

5.3 An assembly with double-pegs and multiple holes 119

5.4 Body b1 and body b4 in one assembled position 120

5.5 Body b1 and body b4 in another assembled position 121

5.6 Body b3 and body b4 in an assembled position 122

5.7 All four bodies in their assembled position — intersection detected 123

5.8 A modified version of the assembly example one 124

5.9 Revised four bodies in their assembled position 125

5.10 Second assembly example: an assembly with all revolute joints 126

5.11 When $b2$ fits to $b3$, the representative points of the fitted features coincide, the orbit of the other representative point forms a locus under this revolute joint motion 127

5.12 Intersecting loci from *compl* of $b1$ and *compl* of $b2$ 128

5.13 An example containing both discrete and continuous joints 129

5.14 A gearbox 130

D.1 Two overlapping lattices 140

G.1 A Chinese phrase: *cheng shang qi xia* meaning: interfacing the upper and the lower. The word *upper* (shang 4), *lower* (xia 4) and *interlock* (ka 3) in Chinese 145

CHAPTER 1

INTRODUCTION

... the two Englishmen are still waiting to be introduced.

— punchline of an old joke

Human beings have long appreciated symmetries of objects. Any subset S of the Euclidean space $\Re^3$, such as a curve, a surface or a solid, has *symmetries*. A *proper symmetry* is a rotation and/or translation, including the identity mapping, which brings S into coincidence with itself. For example, a rotation about the central axis of a cylinder is a proper 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 $\Re^3$, i.e. the cylinder is *setwise invariant* under this rotation. The term *proper* symmetry is used to exclude mirror symmetries (reflections), which although useful for describing the appearance of objects, cannot in general be realized by any physical movement. In this dissertation I shall use the term 'symmetry' to indicate proper symmetry.

In the early nineteenth century, Galois developed an elegant yet powerful mathematical tool for describing symmetry, namely *group theory*. A *group* can be regarded simply as a set of elements together with a binary operation defined on the set that satisfies certain closure properties. Typically, elements in a group are mappings over certain domains. In particular, all the distance and handedness preserving mappings in the Euclidean space, i.e. the so called *rigid motions*[1], form a group under the composition of such mappings. This group is called the *Proper Euclidean group* $\mathcal{E}^+$. The set of all the symmetries of a set S in $\Re^3$ has a group structure and is thus a subgroup of $\mathcal{E}^+$. This group is called the *symmetry group* of S. Symmetry groups can be either finite or infinite, discrete or continuous. Examples of finite groups are the symmetry groups for the platonic solids shown in Figure 1.1 (from [69]).

[1]Once again although reflections are also rigid, they are excluded from the *rigid motions* used in this thesis. Other equivalent terms for rigid motions include: displacements, transformations translations and/or rotations

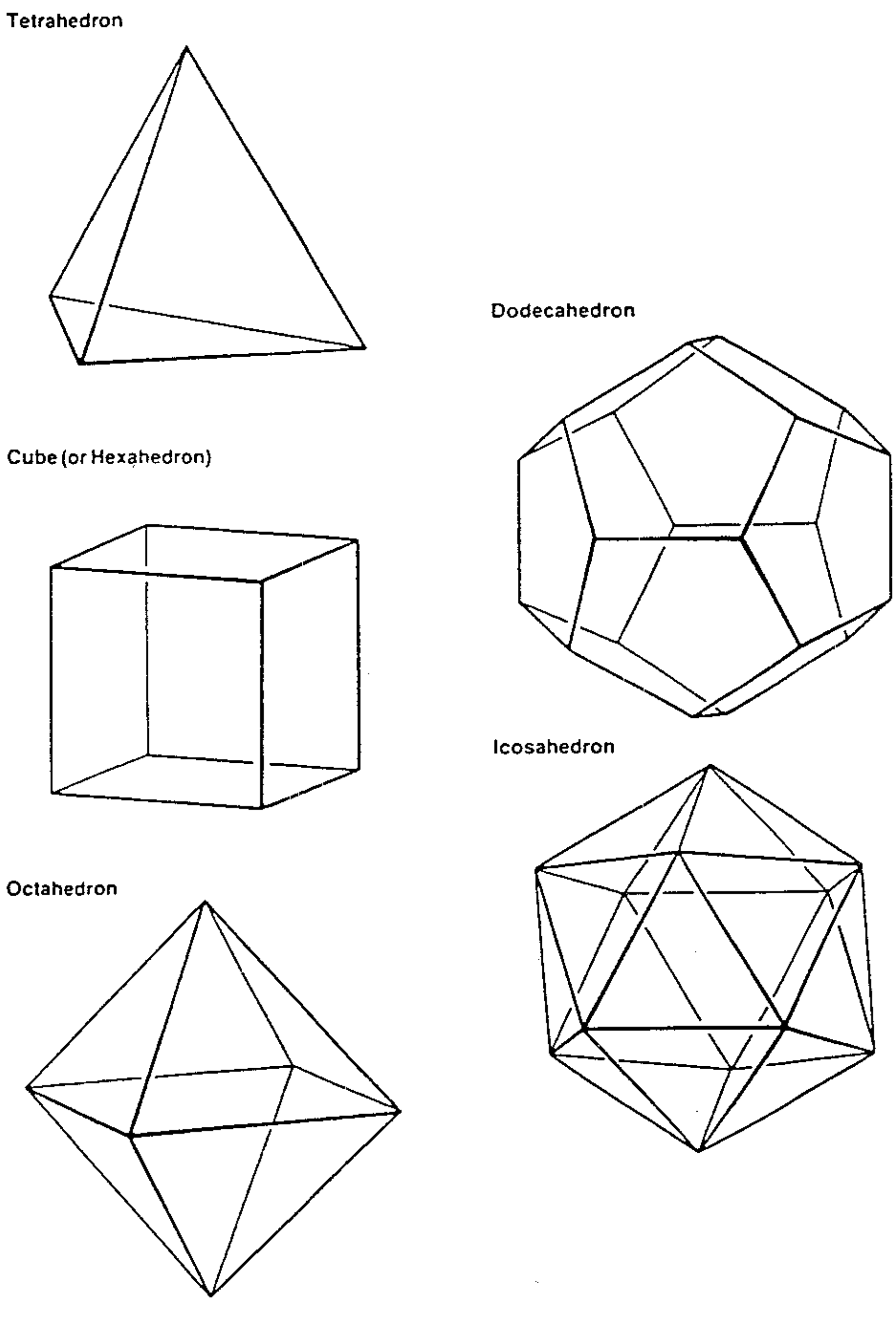

Figure 1.1: Platonic solids

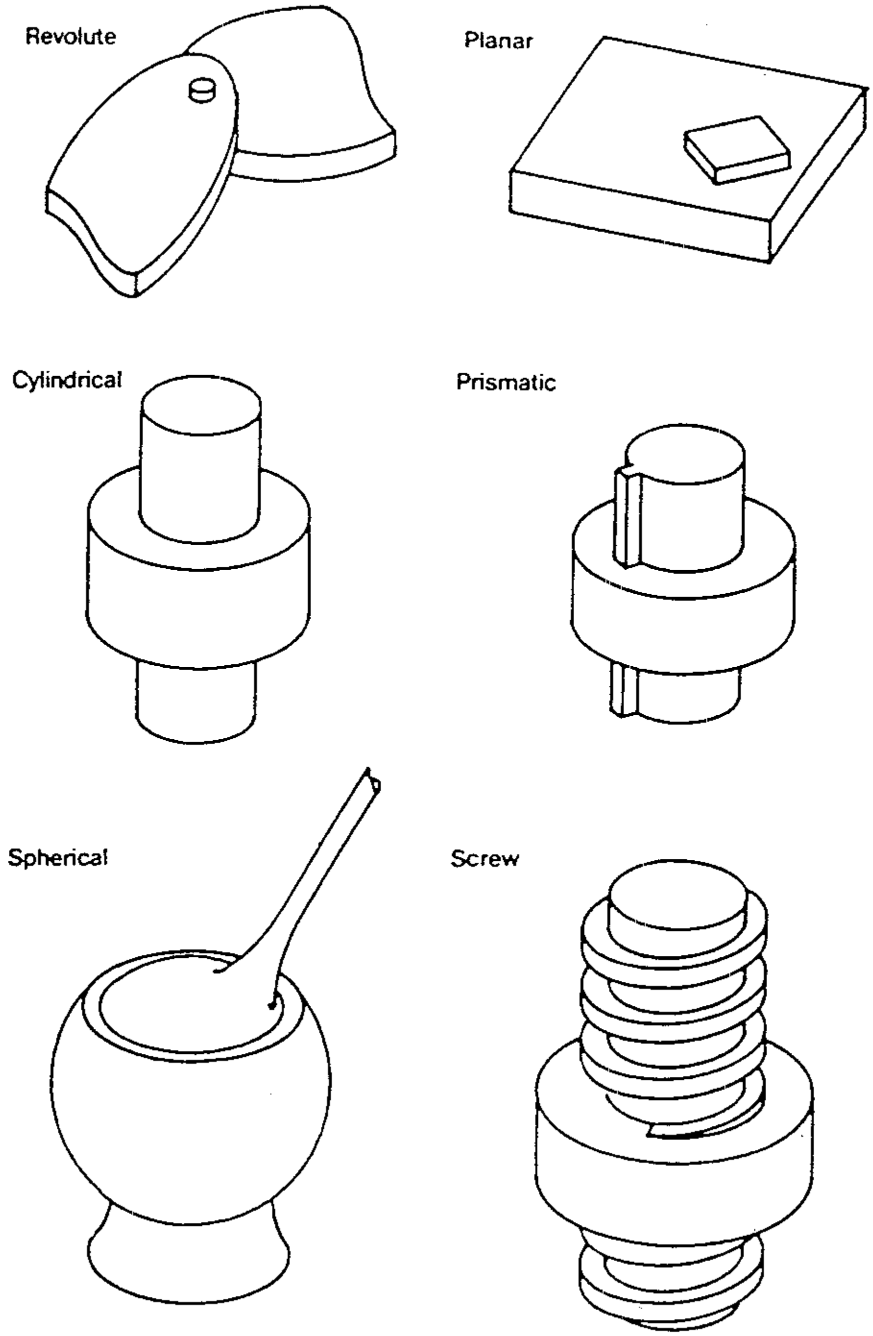

Figure 1.2: Lower Pairs

Examples of continuous groups are illustrated in Figure 1.2 (from [57]). This figure exhibits all six mechanical joints that have surface contact. These joints are called *lower pairs* in mechanical engineering. The essential observation leading to our work is that contacting surfaces from different bodies have the same symmetry group. We assert that not only the symmetries of an isolated body but also the symmetries of certain surfaces of the bodies that are in contact are of great relevance for describing how bodies interact with each other. Such a description is necessary to instruct a robot to accomplish assembly tasks.

In robotic task-level planning [26, 51], 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 robot actions. A robot task-level assembly planner requires, as part of its input, the final assembly configuration to be described [38, 53, 84]. Planning the sequence of assembly also requires the relationships among assembly components to be specified [21, 34, 83], even at the abstract level. However from a mechanical design it is not always trivial to derive an assembly configuration specification that is complete and unambiguous. One element of this problem actually arises from the *symmetries* of assembly components. For example, when using a socket wrench to turn a hexagonal bolt-head, there are six different positions for the socket wrench to be in. Without a proper representation of symmetry a complete specification of this type of contact will be tedious and thus error prone. There is a vital relationship between symmetry and function in mechanical engineering that cannot be dismissed. Therefore, it is desirable to provide an intelligent interface linking the mechanical designer and the robotic task-level planner which 'understands' symmetry and generates task specifications automatically.

Group theory provides the formal treatment of symmetry, and is a basic mathematical tool in physics and theoretical chemistry. It seems natural for today's practitioners concerned with mechanically engineered entities to combine the elegance of group theory with another powerful tool of modern times, the computer, both to

- acknowledge the group theoretical categorization for shapes and the kinematic relationships between rigid bodies; and to
- explore the computational tractability of applying group theory to real problems.

Belinfante and Kolman [3], Hervè [33], and Popplestone [63] are among the few pioneers who contributed to the group theoretical formalization of mechanical engineering practice The work reported in this dissertation extends further along these directions, and serves as an existence proof that such formalization is not only decorative but also functional.

1.1 Statement of the Problem

Robotics is a relatively young and developing field, and standard terminologies have not yet been coined It is therefore necessary to give some explanations of the terms I use in this dissertation.

- *Planning for robotic assembly* is concerned with generating a plan for assembling a product; a plan that will be carried out by robots
- A *plan* is considered to be a detailed assembly task specification For robotic assembly, task specifications must explicitly express the exact final positions and orientations of assembly bodies, for example, via a 4×4 homogeneous transformation matrix Such detailed information is not necessary for manual assembly.
- An assembly task specification consists of two kinds of information:
  1) spatial position/orientation instructions specifying the final or intermediate assembly configuration, i.e. the spatial relationships to be established among certain rigid bodies
  2) temporal instructions on the order of establishing such relationships.
- The robot to execute the plan is expected to be a sensor-based robot that has the ability to make certain decisions contingent on circumstances. Therefore, the output plan is a partially ordered graph instead of a linear sequence, which contains all the possible spatial and temporal choices that are available.

Planning for robotic assembly is a typical problem of *the assignment of values to variables, subject to a set of constraints* [22]. Such constraints may include:

Kinematic constraints: A kinematic constraint specifies a contact between a pair of bodies, albeit one that may not pertain to every situation occurring during a particular assembly. Contacts between surfaces of simple shape can be treated as kinematic pairs [2, 33].

Spatial constraints: No two bodies occupy the same volume of Euclidean space at the same time; such spatial constraints are bounded (in C-space [52]) by kinematic constraints.

Static and dynamic constraints: These relate to the forces acting on bodies during assembly. Sub-assemblies must be stable under gravity, friction must be overcome when necessary, and material must not be overstressed.

Temporal constraints: These constrain the order in which actions may occur. They depend primarily on the previous constraints.

One of the difficulties in robotics research is the interleaving of the application of different constraints. The order in which the constraints are applied is crucial for preserving the completeness and correctness of a planning system. Under a broad assumption that all the assembly tasks will be carried out in Euclidean space, the kinematic and spatial constraints become fundamental. This is so because solutions satisfying these two constraints do not exclude prematurely solutions that will satisfy all the other constraints. It is standard practice in the analysis of mechanical systems to treat the kinematics before the dynamics of the system — indeed the conceptual dependencies and the possibility that the assembly may be performed either on earth or in space demand this. The work described here is concerned with establishing this kinematic basis.

When two bodies in an assembly are in contact, they do not make contact over their whole surface, rather *features* of each body are in contact. A *primitive feature* F of a body in $\Re^3$ is a surface of the Euclidean space $\Re^3$ which is associated with the 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 [87], and captures the idea of features used by such workers as Dixon [23], Henderson [31] , and Green, Carney and Brown [28, 14]. The symmetry group of a body is usually different from the symmetry groups of its features. For example, a cylindrical hole in a block is mapped into itself by any rotation about the central axis of the hole. Such rotations are therefore symmetries of the hole, but not necessarily symmetries of the block (Figure 1.3). Since in an assembly, bodies are related by features, the symmetry of features is more important than the symmetry of the bodies to which the features belong. Such symmetries of features are of essential importance in assembly planning for understanding how features mate and what the final assembly configurations can be.

For realizing the goal of constructing an intelligent task-level planner which understands symmetry, at least the following four sub-tasks need to be accomplished:

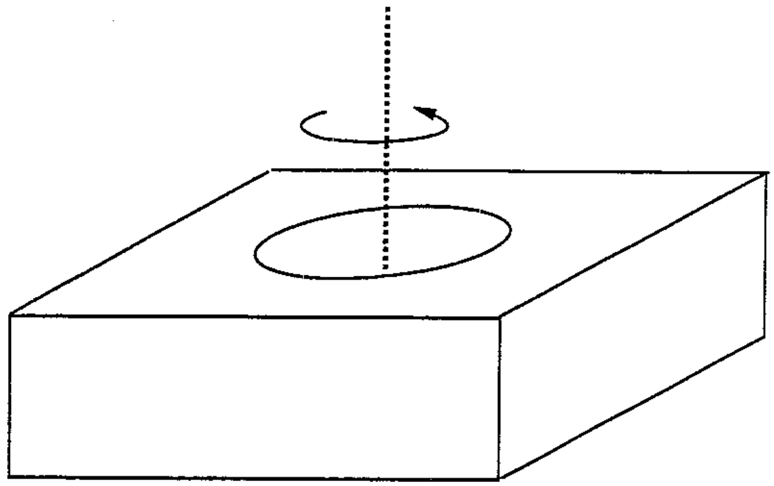

Figure 1.3: A block with a cylindrical hole

1. find a set of candidate mating feature pairs for each kinematic constraint using the boundary models of assembly components and a set of assembly operations between pairs of bodies;

2. find all the possible assembly configurations that satisfy the kinematic and the spatial constraints;

3. find the partially ordered disassembly sequence (PODS) from an assembly configuration for the purpose of finding the sequence of assembly;

4. map the PODS into assembly actions of one or more robot hands under corresponding physical conditions.

The main concerns of this dissertation include sub-tasks one to three which deal with the kinematic and spatial constraints only and are independent of the environment and the robotic hardware used. This set of tasks form, the *kinematic assembly planning* problem[2]. Subtask four will be discussed in Chapter 6.2.

[2]So for now KA3 can be understood as Kinematic Assembly-planning in **3** steps. See Appendix G for its real origin.

The basic beliefs behind this work include:

1) The assembly actions of a robot should be deduced from a thorough understanding of the objects that the robot is going to act upon. Geometric and spatial constraints play an essential role in this decision making process. The geometry of the assembly components enforces some *hard constraints* on the configuration(s) of assembly as well as the order of assembly

2) We treat robotic assembly planning as taking place at two distinct conceptual levels. Planning at the higher level involves deriving *nominal trajectories* along which the bodies to be assembled are to be moved. These trajectories are nominal in the sense that they would accomplish the assembly were we to have a perfect robot manipulating bodies whose shapes were perfectly accurate. Planning at the lower level transforms such a high-level specification into an assembly plan which takes account of uncertainty. In this work we are primarily concerned with discussing high-level robotic assembly planning

We have used the geometric modeller PADL2 [11, 71] (Appendix E) to represent all the assembly components as if they are from a CAD system for mechanical engineering. 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 [53] 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 [74], there is relatively little use made of non-polyhedral models that are geometrically complete. In specifying assemblies to $\mathcal{KA}3$, it is possible to omit details about which body features really mate, stating only that the bodies themselves are to be mated. The sets of possible mating features of two bodies are inferred from their geometric models. This distinguishes $\mathcal{KA}3$ from most robot task-level assembly planning systems, where a detailed, unambiguous assembly task specification is required.

1.2 $\mathcal{KA}3$ is an Interdisciplinary System

The *symmetry group* is an important feature descriptor in this work and has been integrated with the geometric model of each assembly body feature. However, the theory of symmetry groups itself is not sufficient to solve the kinematic assembly-planning problem stated above. Combining inter-disciplinary knowledge in a complementary manner is another characteristic of this work. For example, from a pre-stored salient-feature-library, $\mathcal{KA}3$ finds semantically relevant assembly features from the geometric entities which constitute the boundary model of an object. Such semantically relevant features form the domain of each kinematic constraint. The techniques for solving constraint satisfaction problems, which have been studied extensively in fields such as operations research and artificial intelligence, have been combined in $\mathcal{KA}3$ with symmetry group representation of features and bodies. Reasoning using heuristics to cut down the combinatorics of the search space is also employed. On the other hand, in order to use compound features and to find relative positions of mating bodies, the computation of symmetry group intersections is required, for which a mathematically rigorous treatment of symmetry groups is needed. An efficient group intersection algorithm using characteristic invariants has been developed (Chapter 4). Two related but self-contained aspects have been pursued in this work:

- Representation and intersection of **TR** subgroups of the proper Euclidean group $\mathcal{E}^+$

  I define a geometric representation in terms of *characteristic invariants* for an important family of subgroups of the proper Euclidean group. Each member of this family is called a TR group due to the fact that a TR group is a semidirect product of a translation group T and a rotation group R. Some examples of TR groups are the translation subgroups T^1, T^2, and T^3; the rotation subgroups $SO(2), SO(3), O(2)$, the cyclic groups, the dihedral groups, and the platonic groups; as well as groups containing both rotations and translations such as the symmetry groups of the plane and infinite cylinder, and certain Crystallographic groups. These TR groups and their intersections are very important in describing and analyzing the symmetries and kinematics of assemblies. I prove that there is a one-to-one correspondence between TR groups and their characteristic invariants. I prove that the intersection of TR groups is closed and can be calculated from their characteristic invariants. An efficient intersection algorithm using characteristic invariants has been implemented.

- Combining symmetry groups and AI problem solving techniques into one unified computational framework for assembly planning from solid models

> Being able to represent and reason about symmetries of bodies and features, a high-level assembly planning system $\mathcal{KA}3$ is capable of finding mating features from the geometric boundary models of assembly components and a set of assembly operation requirements between bodies. This permits task specifications to be described in terms of body relationships only. $\mathcal{KA}3$ establishes a constraint satisfaction network (CSN) embedding the kinematic and spatial constraints. For each feasible solution of the CSN (a possible assembly configuration), $\mathcal{KA}3$ applies the translational-disassembly-motion analysis and outputs a partially ordered sequence of contact states of assembly components as a nominal assembly plan for lower-level uncertainty analysis.

The eventual goal is to incorporate this planning work into an integrated robotic assembly system such as Handey (Lozano-Pérez et al 1987)[53], but one that has an intimate comprehension of symmetries not restricted to polyhedral bodies. In this work I consider how a specification for assembly configurations can be determined from a design in a way that exploits the symmetries present in a 3-d body, that is computationally tractable, and that places a minimal load on the human user. The generated task specification is more precise and relevant for lower-level robotic planning than that given by the existing high-level assembly planning systems [21, 34, 83]. This is because it contains complete geometric information of each solid in the assembly, a partially ordered sequence of contact states of mating features, and the relative positions of assembly components. The interaction between algebra and geometry within a group theoretic framework and the interaction between constraint satisfaction problem techniques and heuristic search strategies have provided us with a unified computational treatment for reasoning about how parts with multiple contacting features fit together.

1.3 Organization of the Dissertation

This dissertation is organized into six chapters followed by Appendices and a Bibliography.

Chapter 2 describes related work that has an impact on the work described in this dissertation.

Chapter 3 emphasizes the correspondence between shapes, especially features, and their symmetry groups. In particular, we relate the symmetry group of a compound feature to the symmetry groups of its component primitive features.

Chapter 4 gives a detailed description and justification of the characteristic invariant method for **TR** groups. This chapter starts with some background knowledge and definitions, followed by a series of proofs leading to two main theorems. An efficient group intersection algorithm concludes this chapter.

Chapter 5 is a long chapter in which the whole assembly planning system $\mathcal{KA}3$ is described. This chapter is divided into five sections. Section 5 starts with an overview of the whole system. Section 5.1 illustrates how to find mating features from a set of boundary models of the assembly components. Section 5.2 describes how to use a constraint satisfaction network to apply kinematic and spatial constraints in order to find and verify all the feasible assembly configurations. Section 5.3 is concerned with finding the partially ordered disassembly sequence for each assembly configuration established in Section 5.2, via a translational-motion analysis. Finally, Section 5.4 presents four typical assembly examples, and shows how $\mathcal{KA}3$ solves them.

Chapter 6 assesses the significance of this work, its contributions and limitations, and envisions possible future research based on what has been achieved.

CHAPTER 2

RELATED WORK

. . what's past is prologue . . .

— Shakespeare
The Tempest

The automatic generation of task specifications, as we stated in the Introduction, is related directly or indirectly to many research fields. For example, problem solving and domain independent planning in the field of artificial intelligence, task planning in robotics, process planning in manufacturing and design for assembly in mechanical engineering. In this chapter we review some of this related work to gain a broader perspective of our work.

2.1 Planning in Artificial Intelligence and Planning in Robotics

Traditionally, *planning* in AI is defined as : *Deciding on a course of action before acting* in [16]. Planning has been a central topic in AI approaches to robot programming since the late 1960's. One of the major contributions of AI research is to represent knowledge explicitly and symbolically. Some planning systems in the field of AI, such as STRIPS [25], ABSTRIPS [72], NOAH [73] and MOLGEN [79, 80], have developed important concepts and methods for problem solving which can be incorporated into robotics planning; some examples being non-linear planning, hierarchical planning, meta-planning, problem refinement, constraint posting, and the procedural network representation for plans. However, these AI planning systems lack the ability to represent knowledge about the geometric and physical properties of the task, of the environment and of the robot itself. Moreover, they lack the ability to reason about such knowledge. As Brady points out:

> Robotics challenges AI by forcing it to deal with real objects in the real world. An important part of the challenge is dealing with rich geometric models [9].

Indeed, robotics planning is a complicated field because of vagaries and inexactness in the real world. These complications suggest the need for geometric modeling and spatial reasoning under uncertainty.

2.1.1 Domain Independent Planning in AI

In domain-independent conjunctive planning research, Chapman [15] points out that the nonlinear constraint-posting approach has been most successful in the problem of achieving conjunctive goals. His comments apply to planners such as STRIPS, MOLGEN, NOAH and TWEAK. Having reviewed the conjunctive planners in the literature, Chapman showed that all domain-independent conjunctive planners work the same way. The efficiency and correctness of all these planners depends on the traditional add/delete list representation for actions; efficient (i.e. polynomial time) general purpose planning with more expressive action representations is impossible. The action representation which they depend on is inadequate for real-world planning and desirable extensions to this action representation make planning exponentially hard.

Chapman's result is important for at least the following two reasons:

1. He provided a clean and verifiable summary of AI domain independent planning, and documented its limitations.

2. His work serves as a warning sign to those who would like to use AI planning techniques to solve specific problems. To avoid the inherent intractability of domain independent planning for complex problems, one should perhaps explore special properties of the problem at hand to discover useful, domain specific insights.

The focus of this dissertation is planning for assembly. We will demonstrate that object symmetry is one such domain specific property which enhances the tractability of assembly planning.

2.1.2 Planning in Robotics

One definition for robotics is *the intelligent connection between perception and action* [9]. Approaches which have been taken to achieve such an intelligent connection include: the development of object-level programming languages for robotics, task-level planning and the use of learning. The goal of robot programming is to simplify the process of writing robot programs for complex tasks. A task-level planner is expected to be able to transform task-level specifications in terms of the desired effects on objects, into manipulator-level specifications in terms of robot action commands. One of the earliest task-level robot programming language designs was AUTOPASS[44]. Perhaps the most important aspect of that work was a study of planning collision-free paths among polyhedral objects.

People have different opinions about how planning in robotics is divided. One view centers upon whether a robot is involved in the context of planning. Robotics planning then divides into two different levels:

- object-level planning, which deals with robot independent plans; and
- robot-level planning, where planning concerns a specific robot

In addition, a transition stage maps from one level to the other. Another useful view focuses on the relevant branch of theories involved in different aspects of robotics planning. Here, planning is divided into: kinematic planning[1], dynamic planning and control etc., Work described in this thesis is primarily concerned with the kinematic aspect of assembly planning, which is closely related to the kinematic aspects of both mechanical design and robotics task-level planning.

Handey: A Robotics Task-level Planner

The most complete existing task planning system so far is the MIT *Handey* system of Lozano-Pérez *et al* (1987) [53]. Its domain is simple assembly of planar-faced objects. The input to the system includes: accurate object models for all the parts and a sequence of MOVE commands, each of which specifies an object and its destination. The lesson they learned during this work is : *the need for a systematic and efficient way of dealing with the number of options available while constructing a plan*[53]. *Symmetries* of objects often implies multiple options in assembly tasks; a system like *KA*3 can reduce its search space by exploiting an understanding of symmetries.

[1] *Kinematics* treats motion of bodies apart from aspects of mass and force.

2.1.3 Assembly Planning for Robotics

Assembly Plan Representation

Homem de Mello and Sanderson present a representation for assembly plans based on AND/OR graphs or *hypergraphs* [74]. Each node in such a graph corresponds to an assembly. Those nodes containing only one assembly part are the leaves of the graph. A set of directed arcs related by AND represent a disassembly operation. Each arc points from the original assembly to one of the subassemblies. Any feasible assembly plan corresponds to a tree in the graph. Homem de Mello and Sanderson have proved that a complete and correct set of precedence relationships can be derived from this AND/OR graph. However, the graph of possible disassemblies can be very bushy if symmetries are present in the assembly. A treatment of symmetries could lead to a more compact and efficient representation for assembly plans.

Assembly Sequencing from Precedence Knowledge Acquisition

De Fazio and Whitney have extended Bourjault's work [8] on generating all the possible assembly sequences from a liaison diagram [21]. Although the algorithm for generating all assembly sequences has been implemented successfully [83] it is still unclear how the liaison diagram can be automatically generated and how difficult it is to answer questions asked prior to the generation of assembly sequences. Our work on finding mating features from boundary models of assembly components [48] could be extended to establish liaison diagrams and answer questions based on the geometric, spatial and kinematic constraints, thereby complementing to De Fazio and Whitney's work.

2.2 Mechanical Design and Automatic Assembly

2.2.1 Design for Assembly

The attention of researchers has been drawn to the application of AI techniques in the design-to-manufacturing spectrum. This has helped to bring manufacturing knowledge to bear early on in the design process. On the other hand, AI knowledge representation prototypes are enriched by requiring design geometry to be well presented. As Dixon and Dym point out: *geometry is a crucial issue of knowledge representation linking design with manufacturing* [23]. They propose representing design geometry in terms of features, where a *feature* of a design is in an identifiable geometric shape of an entity. They reason that *most of the knowledge needed for evaluation of manufacturability, process design, and process planning is expressed in terms of features and combinations of features* [23]. Our work explores these possibilities, and KA3 can extract information about the features from a boundary representation, or can use features present at the very beginning of a design [23].

2.2.2 Automatic Assembly

Boothroyd, Poli and Murch address automatic assembly problems in [7]. Automatic assembly is different from robotic assembly in that less versatile devices are used. They observed that in addition to physical laws such as gravity, geometry plays an important role in automatic assembly, with the symmetry properties of parts being especially relevant. In [5, 7, 6] α and β symmetries are defined in order to map the rotational symmetries of an assembly part to a coding system that can be used to look up the assembly cost. In this way, the cost of assembly can be minimized during the design stage.

Boothroyd et al define α-symmetry to be: *the rotational symmetry of a part about an axis perpendicular to the axis of insertion.* Likewise, β-symmetry is *the rotational symmetry of a part about its axis of insertion. The magnitude of rotational symmetry is the smallest angle through which the part can be rotated and repeat its orientation* [7]. This definition has its limits. First of all, the insertion axis has to be identified. When more than one insertion axis is present, as in a tetrahedron for example, the characterization has to use more than one pair of α-β symmetries. Secondly, α-β symmetries are defined for parts(bodies) only, but features are actually more important in an assembly. In [47] these definitions were used to describe the symmetries of both bodies and features for an assembly planning system.

Boothroyd et al's work has provided one of the practical justifications for our concern with symmetry. Our feature symmetry group representation is a generalization of α-β symmetries that is placed in the coherent mathematical framework of group theory.

2.3 Geometric Reasoning

2.3.1 From Spatial Relationships to Relative Positions

RAPT by Popplestone, Ambler and Bellos [1, 65] is a software system which infers the positions of bodies from specified symbolic spatial relationships. Its implementation is based upon constraint propagation and equation solving. In RAPT, world states are defined in terms of the relationships between features of bodies, as opposed to relationships between bodies in our work. Our work differs from that of RAPT in two ways: one is the incomplete kinematic constraint input (body relationships can be given instead of feature relationships); the other is that instead of mapping symbolic representation for spatial relationships into a set of algebraic equations, symmetry groups of the mating feature pairs are generated first [50, 67, 63]. This provides greater computational tractability and a more uniform treatment of spatial relationships, plus the ability to treat multiple feature relationships through symmetry group intersections. It also treats finite symmetries which RAPT does not treat.

Both RAPT and ACRONYM [10] used results from group theory in the simplification of their equalities and inequalities. Work described in this thesis is using relevant group theory to avoid going into the process of symbolic equation manipulation altogether.

2.3.2 Feature Inferencing

In his Ph.D. thesis work [31] Henderson studied the extraction of feature information from three dimensional CAD data. The purpose of his work was to extract manufacturing-specific semantic knowledge about the part. The procedure consisted of searching the part description, recognizing cavity features, extracting those features as solid volumes of material to be removed and arranging them in a feature graph.

In [28, 14] Green, Brown and Carney investigate the relationship between *shape* and *fit* (in this case, direct insertion). They divide the feature inferencing process into five layers:

- grouping: group features according to different attributes,
- topology: recognize patterns,
- orientation: align surfaces between two objects,
- matching: match feature pairs, and
- confirmation: verify the compatibility of matched feature-pairs, volume, height and attached surface area.

To date, our work is unique in its use of a uniform representation such as the symmetry group as a descriptor for feature characteristics, and in its efficient computational mechanisms for combining features' symmetry groups.

2.3.3 Degree of Freedom Analysis

Possible Motion Analysis

Jakopac describes a method for automatically deducing the kinematics of certain mechanisms from their solid model description, and considers how to find possible motions given a set of *joint face pairs* [39]. A *joint face pair* (JFP) is a pair of faces from two separate bodies that overlap and have opposing normals at all points of overlap. Figure 1.2 shows all the *lower-pair joints* — those having only surface contact between links. Among these only planar, cylindrical and spherical JFPs were considered, and the examples given in his thesis contain only planar and cylindrical pairs. He considered only those kinds of motion that allow contacting faces of bodies to maintain contact at the next instant of time. He noticed that of all the kinds of lower pairs, only planar, cylindrical, spherical and screw pairs can be easily identified since they have only one JFP.

Jakopac's work is interesting in that it permits enhancement of the motion analysis method by introducing the symmetry group representation in mechanical links. This method can be used to analyze how to disassemble a *mechanism*, i.e. an assembly of rigid bodies with surface contact only.

2.4 General Motion Planning versus Contact Motion Analysis

Roughly speaking, there have been two major approaches to motion planning. *Local methods* such as artificial potential field methods can be fast but are not guaranteed to find a path (i.e. they are incomplete). *Global methods*, like the Roadmap Algorithm presented by Canny [13], are guaranteed to find a path but may spend exponential time doing it.

From the point of view of assembly, we are concerned most with motions of bodies that are in contact, instead of avoiding collision entirely. After all, an assembly process is a process of controlled collision. Hopcroft and Wilfong [37] have shown that, for many trajectory planning problems, any path is homotopically equivalent to a path in which the moving object is in contact with one of the obstacles. If

Table 2.1: Continuous Groups and their Degrees of Freedom

Dimension(d.o.f.)	Group (constraint)	Associated Lower pair
0	1	none
1	T^1	*Prismatic*
1	$SO2$	*Revolute*
1	$\mathcal{G}_{screw}$	*Screw*
2	T^2	none
2	$\mathcal{G}_{cylinder}$	*Cylindrical*
3	T^3	none
3	$\mathcal{G}_{plane}$	*Planar*
3	$SO3$	*Spherical*
3	$T^2 * \mathcal{G}_{screw}$	none
4	$T^3 * SO2$	none
6	$\mathcal{E}^+$	none

two bodies are in contact then some relationship must hold between body features. This was exploited by Koutsou [41], who made use of the RAPT inference engine to search for paths in contact. In her Ph.D. thesis "Planning Motion in Contact to Achieve Parts Mating" [41] she studied how to move a 3-dimensional polyhedral object while maintaining contact with a set of stationary obstacles. She created all path segments characterised as those conjunctions of spatial relationships which restricted the related bodies to have one degree of freedom of motion. Each path segment was terminated by a vertex obtained by conjoining an additional relationship. The ROBMOD modeller [12] was used to determine which vertices were legal, and hence which path segments were legal. Extended legal paths were then made up of concatenated legal path segments.

This approach does in fact find a number of different paths in the same homotopy class. However, as formulated by Koutsou, it is also incomplete and of high complexity in the number of body features. Without an optimization stage, paths are found that are more suitable for fine motion planning than gross motion planning, where it is often desirable to preserve a certain distance from obstacles.

Laugier has studied fine motion planning in terms of the geometry of contact surfaces between bodies [43]. His work is interesting because it introduces a computational representation for directions of motions, and a fast analysis of the effects of a motion on the spatial relationships of bodies in contact. This is very useful when combined with our symmetry group representation for features. Contacting faces of one body with the rest of the assembly form a compound feature which can be used to predict the next possible motions during disassembly analysis.

2.5 Applying Group theory in Robotics

While kinematic and geometric problems are common in robotics, very few papers can be found in the literature that use group theory as an analytical tool.

2.5.1 Group Theory and Mechanisms

Hervé has introduced a rational classification of mechanisms by applying the theory of continuous groups. Since each lower-pair allows a set of relative motions of the two coupled bodies, these motions can be regarded as subgroups of $\mathcal{E}^+$. The number of independent variables required to define the relative position of two coupled links is referred to as their *degree of freedom*. The concept of degree of freedom of a kinematic pair can be extended to a subgroup of $\mathcal{E}^+$ in 3-space, the corresponding concept being that of *dimension*. Table 2.1 is adapted from [33]; for the definition of each group see Table 3.1. This table shows the degrees of freedom (dimension) for some of the continuous subgroups of $\mathcal{E}^+$, and the associated lower-pairs if any.

Hervé represented the intersection and composition of constraints in terms of groups. If there are two relations R_1, R_2 between $body_1$ and $body_2$ then the conjoined relation of $body_1, body_2$ is $R_1 \cap R_2$. When relations are composed one has the following relationship between the dimensions:

$$dim(R(i,k)) = dim(R(i,j)) + dim(R(j,k)) - dim(R(i,j) \cap R(j,k)) \qquad (2.1)$$

where i, j, k refer to three distinct bodies. Equation (2.1) shows the usefulness of subgroup intersections. A look-up table was given in [33] that contains only intersections of the continuous subgroups of $\mathcal{E}^+$ under different cases.

2.5.2 Group Theory and Robotics

In [63] Popplestone relates robotics and group theory by pointing out:

- The symmetry group of a **feature** is a useful descriptor, perhaps even more important than the symmetry group of a body.

- Features can be considered as infinite surfaces.

- Not only continuous groups (as previously used in describing kinematic joints) but also finite and discrete groups should be handled.
- The family of groups G that can be written as $H_{r_i}H_{t_i}$ where H_{r_i} is a rotation subgroup and H_{t_i} is a translation subgroup is of importance in robotics. This initiated our current formulation of **TR** groups.
- Spatial relations can be described in terms of cosets and the intersections of them, the result can be either a coset or null.
- The advantage of this formulation is to avoid combinatorics from multiple relationships.

Our work extends many of these ideas. In particular, we have advanced the representation of a well-defined family of subgroups of $\mathcal{E}^+$, namely the **TR** groups, to make efficient group intersection possible.

Thomas and Torras [81] used ideas from [1, 33, 63, 65] to find configurations of a set of rigid bodies from a given set of d.o.f. constraints on the bodies. A **constraint** between two bodies is the chain product of a symbolic 4×4 matrix which is pre- or post-multiplied by constant displacements, i.e. a two sided coset. Therefore the problem of finding values for the variables (d.o.f.) associated with constraints can be reduced to the problem of obtaining the cycles appearing in a directed graph whose nodes are bodies and whose arcs are constraints, and solving their corresponding matrix equations. Thomas and Torras tabulate the outcomes of intersection and multiplication of certain continuous subgroups of $\mathcal{E}^+$ (based on tables from Hervé [33]). In their algorithm, the tables are used to simplify certain algebraic equations. In that sense it is simpler than using a large number of rewriting rules [1].

This work is a welcome effort in combining the group theoretic formulation of continuous groups developed by Hervè with the RAPT formation of constraints and equation solving approach. As in RAPT, they still need to use symbolic manipulations, which can be very slow. They also do not treat discrete or finite symmetry groups[2].

2.5.3 Computational Treatment of Groups

Due to the fact that symbolic algebra is typically a costly computation, Popplestone, Weiss and Liu [68] proposed a new approach to deal with group intersections. An

[2]The first two assembly examples using continuous symmetry groups in Section 5.4 are adopted from their papers.

Table 2.2: Characteristic Invariants of Important Subgroups of $\mathcal{E}^+$

Group	Characteristic	Invariants
G	Rotational(G/$\mathcal{T}^3$)	Translational
1	Vectors $r_1 r_2$	Point
$\mathcal{T}^1$	Vectors $r_1 r_2$	Line $l_1 \parallel r_1$
$\mathcal{T}^2$	Vectors $r_1 r_2$	Plane with normal r_1
$\mathcal{T}^3$	Vectors $r_1 r_2$	$\emptyset$
SO(2)	Vector r_1	Point
$\vec{cylgr}$	Vector r_1	Line $l_1 \parallel r_1$
$\vec{\mathcal{G}}_{plane}$	Vector r_1	Plane with normal r_1
$SO(3)$	$\emptyset$	Point

invariant S of a group G is defined to be a subset of $\Re^3$ such that for all $g \in G, g(S) = S$. An invariant so defined is not unique for a group G, however. As stated in [68], *we can use a unique set of characteristic invariants as a way of characterising some important subgroups of the Euclidean Group. This makes it possible to start with two symmetry groups, compute their characteristic invariants, combine these invariants, and then recover the intersection of the original groups from the combined invariants.* The set of groups treated in [68] and their invariants are listed in Table 2.2.

Because the choice of characteristic invariants had a certain arbitrariness, it led to difficulties in extending that work beyond a limited class of groups. In Chapter 4 of this dissertation, I redefine characteristic invariants, to give us the power of characterizing an important family of subgroups of $\mathcal{E}^+$, i.e. those groups that can be written as the product of a translation subgroup T and a rotation subgroup R, as suggested in [63]. Naturally, this family of subgroups is called the **TR** subgroup of $\mathcal{E}^+$. There are an infinite number of **TR** groups including continuous as well as discrete groups, infinite as well as finite ones. A proper subset of **TR** groups called *tractable groups* excluding the discrete translation subgroups is treated in [88]. Our approach maintains the spirit algebraic problem solving, while using a geometric methodology that provides a set of relatively intuitive proofs for the justification of characteristic invariants and the closure property of **TR** groups (Chapter 4, [46]), as well as a computationally efficient algorithm for **TR** group intersections.

CHAPTER 3

FEATURES, SYMMETRIES AND SYMMETRY GROUPS

Tyger, Tyger burning bright,
In the forests of the night:
What immortal hand or eye,
Dare frame thy fearful symmetry?

William Blake

In this chapter we shall examine the relationships between shapes and their symmetry groups. We shall start with some definitions for abstract groups, then restrict our focus to the proper Euclidean group through the rest of the chapter. See Appendix B for a review of some basic topology definitions used in this chapter

3.1 Abstract Groups and the Proper Euclidean Group

Definition 3.1
A **group** *G is a set of elements under an associative binary operation $\otimes$ satisfying*

- *G is closed under $\otimes$: that is for all $g_1, g_2 \in G, g_1 \otimes g_2 \in G$;*
- *G has an identity element 1_G, for all $g \in G, 1_G \otimes g = g \otimes 1_G = g$;*
- *Each member g of G has an inverse $g^{-1} \in G$ obeying: $g \otimes g^{-1} = g^{-1} \otimes g = 1_G$*

Henceforward we shall omit $\otimes$, e.g. $g_1 g_2 = g_1 \otimes g_2$

Definition 3.2 *A* **subgroup** *H of G is a subset of G that is itself a group.*

Definition 3.3 *The two subgroups $H_1, H_2 \subset G$ are said to be* **conjugate** *if $H_2 = aH_1a^{-1}$ for some $a \in G$.*

Definition 3.4 *If H is a subgroup of G, the* **conjugation class of subgroup** H *consists all the subgroups $H' \subset G$ such that there exists $g \in G, H = gH'g^{-1}$.*

Definition 3.5 *For any element g in G, $gH = \{gh | h \in H\}$ is a* **left coset** *and similarly, 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.*

Note, a coset is not in general a group.

Let $\Re^3$ be three-dimensional real Euclidean space, $x = (x_1, x_2, x_3), y = (y_1, y_2, y_3) \in \Re^3$.

Definition 3.6 *The bilinear form*

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

defines an **inner product** *on this space.*

Definition 3.7 *The inner product defines a* **norm** *for any point of $\Re^3$*

$$||x|| =< x, x >^{1/2}$$

often called the **Euclidean length** *of $x \in \Re^3$, and a metric,* **dist** *for any pair of points of $\Re^3$*

$$\mathbf{dist}(x, y) = \|x - y\| =< x, y >^{1/2}.$$

Definition 3.8 *A map $g : \Re^3 \to \Re^3$ is an* **isometry** *if and only if $\forall x, y \in \Re^3$*

$$||g(x) - g(y)|| = ||x - y||.$$

In other words, an isometry is a *distance preserving* mapping or *rigid transformation*.

Definition 3.9 *The set of all the isometries of $\Re^3$ forms a group with the composition of isometries as group product; this group is called the* **Euclidean group**.

Elements of $\mathcal{E}$ are rotations, translations, reflections and combinations of these.

Definition 3.10 *If we restrict the set of isometries to those isometries which preserve handedness, thus excluding reflections, then the remaining set still has a group structure and is called the* **Proper Euclidean group**, *denoted by $\mathcal{E}^+$.*

Any theory of robotic assembly must consider how proper rigid motions imposed by robots act on $3D$ bodies to realize certain spatial relationships. Thus **the proper Euclidean group** $\mathcal{E}^+$ and its subgroups are of great relevance.

3.2 Primitive Features and Their Symmetry Groups

In this section we give a formal definition for features. After defining the symmetries of a feature, we shall 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 3.11 *A* **primitive feature** F *is a connected, irreducible and non-trivial algebraic surface of* $\Re^3$

In this section we treat primitive features as subsets in $\Re^3$ We shall discuss oriented features in Section 3.5. In practice we use a repertoire of primitive features provided by the geometric solid modeller PADL2 [11] in its data-structures for the boundary model of a solid. Examples include the infinite plane, infinite cylinder, cone, sphere and torus. A primitive feature may contain one or more bounded faces of a solid.

For a set of points S in the Euclidean space we define its symmetries as:

Definition 3.12 *An isometry* $g \in \mathcal{E}^+$ *is a* **proper symmetry of a set** S *if and only if* $g(S) = S$.

Proposition 3.13 *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 3.1) G is a subgroup of $\mathcal{E}^+$ □

Definition 3.14 *When* S *is a feature, the above group* G *associated with* S *is called the* **symmetry group of the feature** S.

Definition 3.15 *Two primitive features* F_1, F_2 *are said to be*

- **distinct** if for all the open subsets[1] $S_1' \subset F_1, S_2' \subset F_2$, no $g \in \mathcal{E}^+$ exists such that $g(S_1') \subset F_2$ or $g(S_2') \subset F_1$. See Figure 3.1 for an example of two distinct features.

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

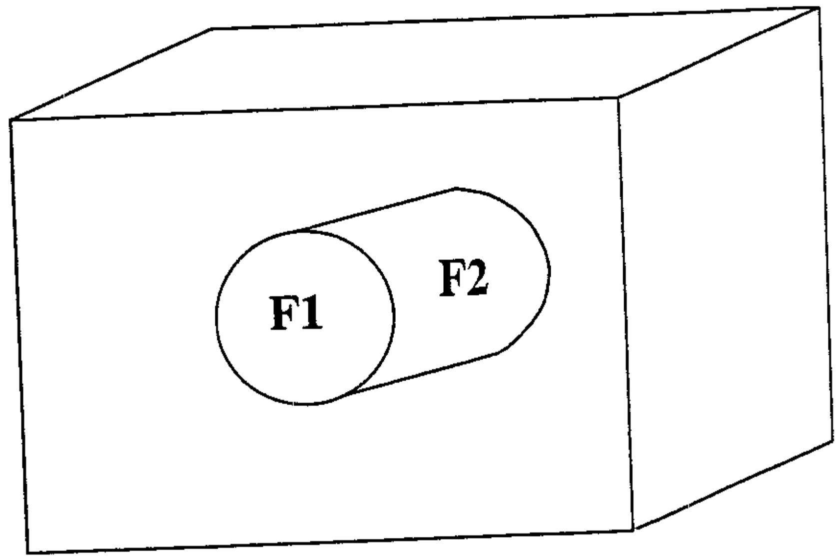

Figure 3.1: A pair of **distinct** features F_1, F_2

- **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 $g, g(F_2) \neq F_1$ simultaneously. For an example see Figure 3.2.
- **2-congruent (strongly-congruent)** via g_c if there exists $g_c \in \mathcal{E}^+$ such that $g_c(F_1) = F_2$ and $g_c(F_2) = F_1$. For an example see Figure 3.3.

The following proposition shows how features and their symmetry groups are related.

Proposition 3.16 *If G is the symmetry group of S 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 $H = H_a$.

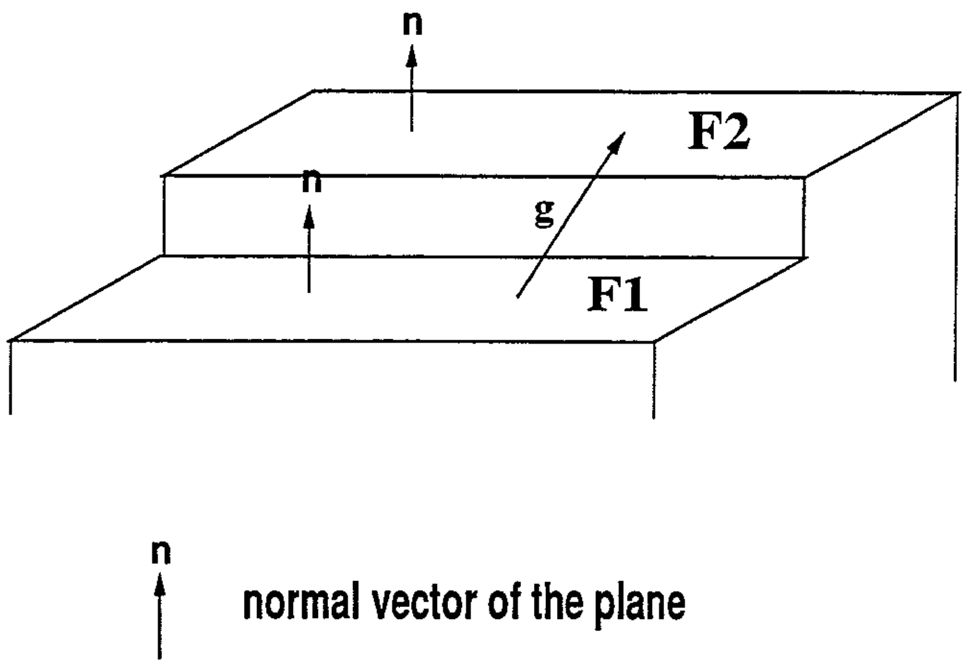

Figure 3.2: Two directed plane features F_1, F_2 which are **weakly congruent** to each other

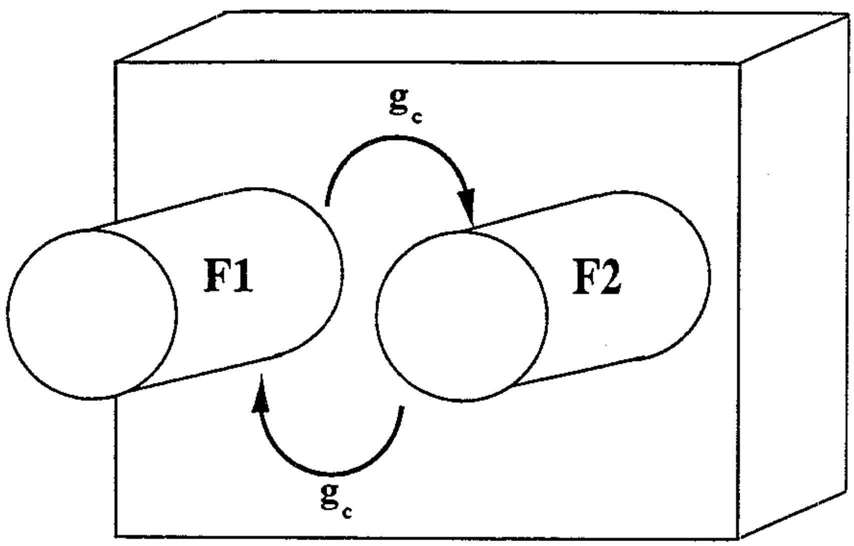

Figure 3.3: Two cylindrical features F_1, F_2 which are **strongly congruent** to each other

Table 3.1: Some important subgroups of $\mathcal{E}^+$

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

By Proposition 3.16, when a feature is relocated by a transformation a, the symmetry group of the relocated feature will be the conjugation by a of the symmetry group of the original feature. This suggests that a conjugation class of a subgroup can be represented in a compact way, that is to represent any feature symmetry group by making it a conjugate of **a canonical symmetry group**. There is one canonical symmetry group in each conjugation class of a symmetry group in $\mathcal{E}^+$. 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. A list of some important canonical subgroups of $\mathcal{E}^+$ with their definitions is given in Table 3.1. Figure 3.4 shows subgroup relationships between some important subgroups of $\mathcal{E}^+$. The arrow $\mathcal{G}_1 \rightarrow \mathcal{G}_2$ in Figure 3.4 means that $\mathcal{G}_2$ is a subgroup of $\mathcal{G}_1$. Table 3.2 gives some of the correspondences between subsets of $\Re^3$ and their canonical symmetry groups.

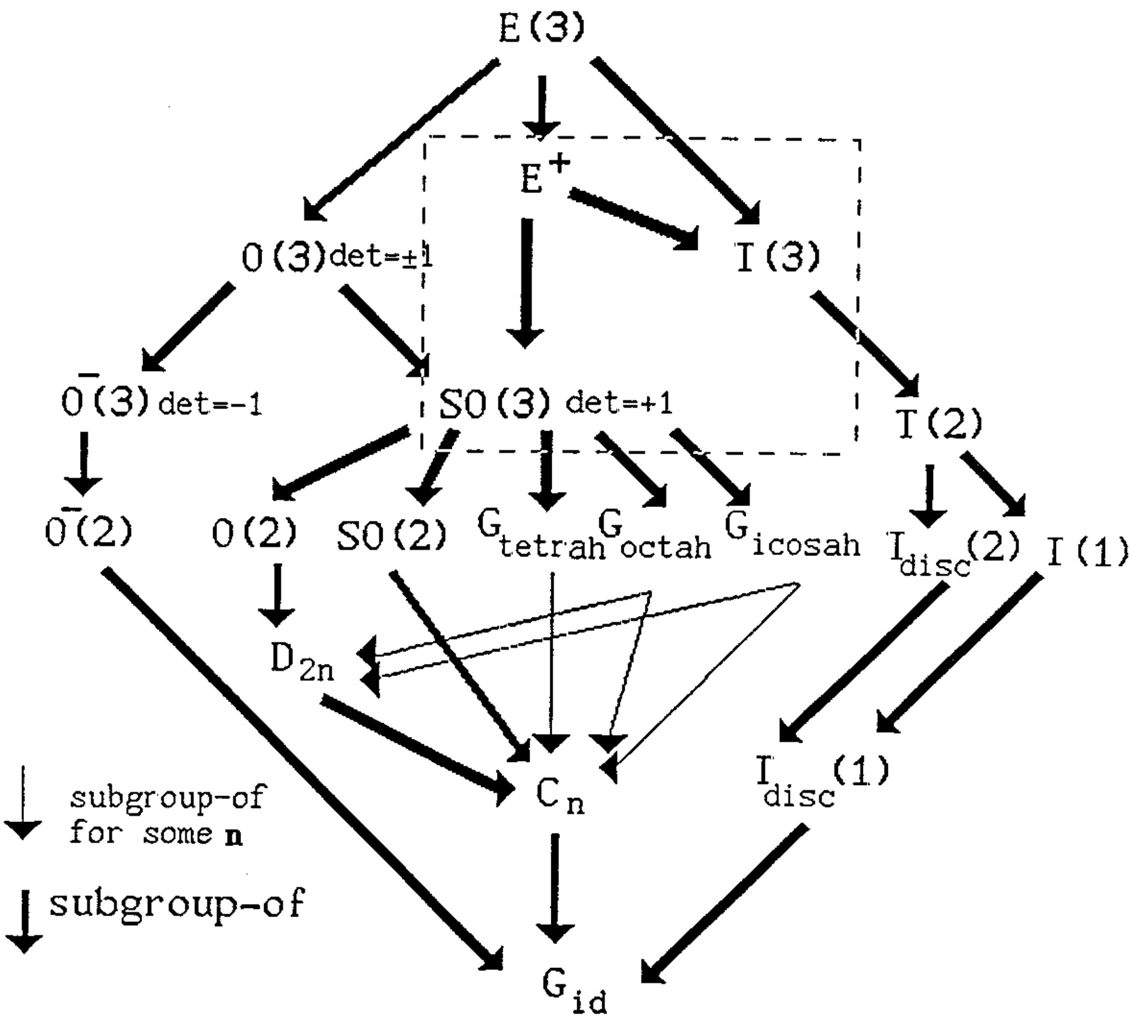

Figure 3.4: Relationships among some important subgroups of $\mathcal{E}^+$

Table 3.2: Correspondence between shapes and their symmetry groups

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

3.3 Compound Features and Their Symmetry Groups

Now let us formally define a compound feature.

Definition 3.17 *A* **compound feature** *F is a set union of n primitive features $F_1, ..., F_n$, i.e. $F = F_1 \cup ... \cup F_n, n > 1$.*

In the next few propositions we shall explore how the symmetry group of a compound feature is related to the symmetry groups of its component primitive features.

Proposition 3.18 *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 $g(F_1 \cup ... \cup F_n) = g(F_1) \cup ... \cup g(F_n) = F_1 \cup ... \cup F_n$. Then $g(F_i) \subseteq F_1 \cup ... \cup F_n$. Suppose $g(F_i) \not\subseteq F_i$. Then there exists a point $y \in g(F_i)$ and for some $j \neq i, y \in F_j - F_i$.

Now let ϵ be the distance between y and F_i. Since F_i is closed, $\epsilon > 0$. Since $y \in g(F_i)$, there must exist $x \in F_i$ such that $y = g(x)$ (Figure 3.5).

Let $O_x \subset F_i$ be a neighborhood of x with radius equal to ϵ. There must exist a point x_1 such that $x_1 \in O_x$ and $x_1 \notin F_q$ for some $q \neq i$, since if no such point x_1 exists then F_i and F_q share the open set O_x and thus are not distinct. Now take a neighborhood O_{x_1} of x_1 and repeat the same argument. Since there are a finite number of features, $F_1, ..., F_n$, we shall eventually find a point x_0 and its neighborhood O contained **only** in F_i, i.e. $O \subset O_x \subset F_i$ and $O \not\subset F_j$ for $j \neq i$ (Figure 3.5).

Since g is an isometry (a distance preserving map), every point of O must be mapped by g to another point within distance ϵ of the point $g(x) = y \in F_j$; but the point is not necessarily mapped to F_j. Clearly, $g(O) \cap F_i = \emptyset$. Suppose a point $p \in O$ has a neighborhood in O with radius ϵ_0. Then p is mapped by g to some feature $F_k, k \neq i$, i.e. $g(p) \in F_k$. Then $g(p)$ is contained in a neighborhood O_k of F_k, let the radius of O_k be less than $\epsilon_0/2$. Since g is an isometry, every point of O_k must be mapped by g^{-1} to a point within distance $\epsilon_0/2$ of point p thus it has to be contained in $O \subset F_i$. Hence $g^{-1}(O_k) \subset F_i$. So F_i and F_k are not distinct, a contradiction.

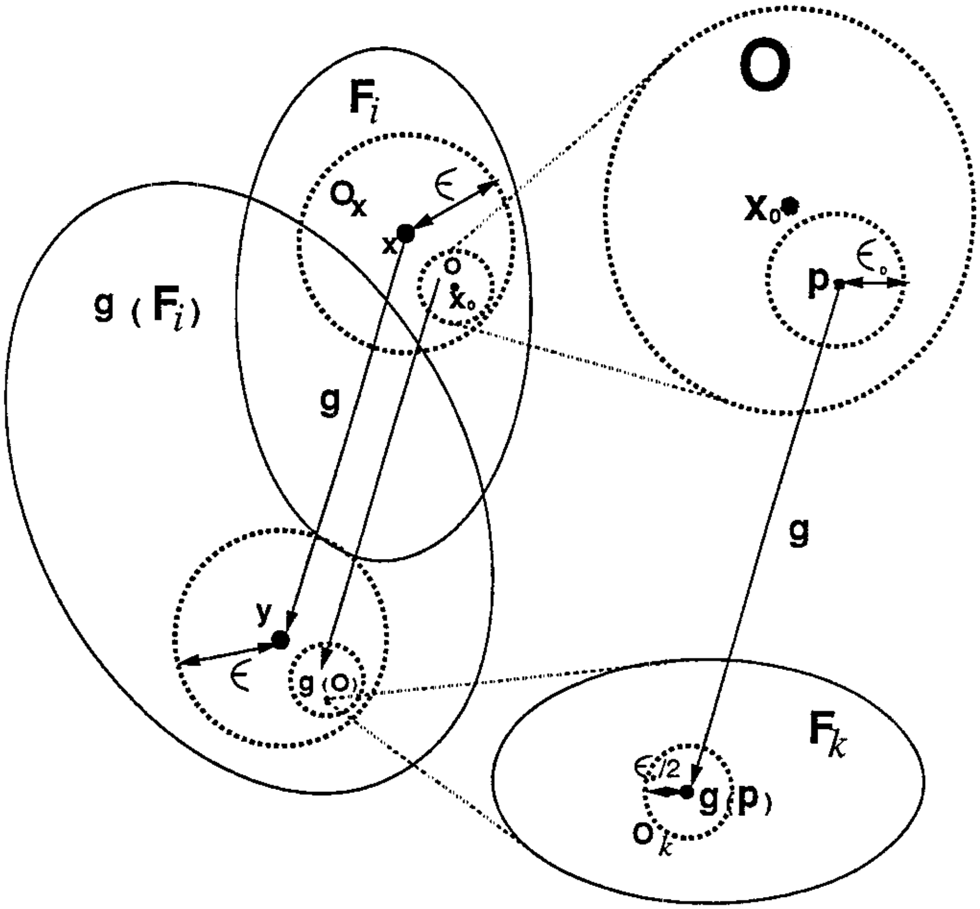

Figure 3.5: Distinct features and relations among their open sets

Therefore $g(F_i) \subseteq F_i$. Since g is a bijection, $g(F_i) = F_i \Rightarrow g \in G_i$ for $i = 1, \ldots, n$. Thus $g \in G_1 \cap \ldots \cap G_n \Rightarrow G \subseteq G_1 \cap \ldots \cap G_n$.

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

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

Lemma 3.19 *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 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 B.2). By Theorem B.3 (see Appendix B), $g(F_1) \subseteq F_1$ 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 2-congruent by definition, a contradiction. Therefore $g(F_1) = F_1$ and $g(F_2) = F_2$. □

Proposition 3.20 *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 3.19, 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 3.21 *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 B.3 (Appendix B), $g(F_1) \subseteq F_1$ 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 3.22 *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 subgroup of $\mathcal{E}^+$ generated from g_c.*

Proof:

If $g \in G$ then by Lemma 3.21 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 $G =< g_c > (G_1 \cap G_2)$. □

This proposition can be extended to n-congruence, i.e. to calculate the symmetry group for n features of a body that permute among themselves When features of one body are strongly congruent with each other, more symmetries are generated than exist before the features are considered collectively.

3.4 Symmetry Groups at Areas of Contact

An assembly is *a collection of bodies which are connected by their features*. When two bodies are connected, they do not make contact over their whole surface, rather certain *features* of each body are in contact. Therefore, although the symmetries of a body affect the final assembly configurations, the symmetries of the features in contact play a much more crucial role.

Let B_1 and B_2 be two bodies, with primitive features F_1 and F_2 which are in contact and have symmetry groups $sym(F_1), sym(F_2)$ respectively. Suppose l_1, l_2 specify the locations of bodies B_1, B_2 in the world coordinate system and f_1 and f_2 specify the locations of features F_1, F_2 in their respective body coordinates. Consider what we can infer about the *relative location* of two bodies that have two features in contact. There are three possible kinds of contacts between B_1 and B_2, they are *point* contact, *line* contact and *surface* or *areal* contact. Regardless of what kind of contacts occur between F_1 and F_2, by the definition of symmetry groups, it is clear that if we move B_1 or B_2 by a member of $sym(F_1)$ or $sym(F_2)$ respectively, the relationship between the features is preserved. A spatial relation between two bodies in contact is a binary relation $\tau \subset \mathcal{E}^+ \times \mathcal{E}^+$, where each pair $(l_1, l_2) \in \tau$ specifies one pair of possible positions for B_1 and B_2. In particular, when the two bodies have an areal contact via F_1, F_2, i.e. a *fitting* relationship, the contacting features coincide and thus their symmetry groups are identical. The spatial relationship then can be expressed as:

$$\tau = \{(l_1, l_2) | l_1^{-1} l_2 \in f_1 G f_2\} \tag{3.1}$$

where $G = sym(F_1) = sym(F_2)$

The relative location of body B_1 with respect to body B_2 can be expressed as:

$$l_1^{-1} l_2 \in f_1 G f_2^{-1} \tag{3.2}$$

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.* This coset is an infinite set when the symmetry group is of infinite order.

Two bodies in an assembly are typically related to each other through multiple primitive features. If the above two bodies are related by *fitting* two pairs of features, i.e. F_{11} fits F_{21} and F_{12} fits F_{22} with feature locations in their body coordinate systems $f_{11}, f_{21}, f_{12}, f_{22}$ correspondingly, then the relative location of body B_1 to body B_2 should be constrained by both relations expressed in the form (3.2) *simultaneously*. That is:

$$l_1^{-1} l_2 \in f_{11} G_1 f_{21}^{-1} \bigcap f_{12} G_2 f_{22}^{-1} \tag{3.3}$$

i.e. a member of the intersection of two two-sided cosets, where $G_1 = sym(F_{11}), G_2 = sym(F_{12})$.

Since each two-sided coset can be rewritten as a one-sided coset as follows

$$g_1 H g_2 = g_1 H g_1^{-1} (g_1 g_2) \tag{3.4}$$

where $H \subset G, g_1, g_2 \in G$, we can modify (3.3) into the format of the intersection of two one-sided cosets.

$$l_1^{-1} l_2 \in (f_{11} G_1 f_{11}^{-1}) f_{11} f_{21}^{-1} \bigcap (f_{12} G_2 f_{12}^{-1}) f_{12} f_{22}^{-1} = (G_1' \bigcap G_2' f') f'' \tag{3.5}$$

where $G_1' = f_{11}G_1f_{11}^{-1}, G_2' = f_{12}G_2f_{12}^{-1}, f' = f_{12}f_{22}^{-1}f_{21}f_{11}^{-1}, f'' = f_{11}f_{21}^{-1}$. Further from proposition 2 of [63] :

If H_1 and H_2 are subgroups of G and $g \in G$, then $H_1 \cap H_2g$ is either null or is a coset of $H_1 \cap H_2$.

we have

$$l_1^{-1}l_2 \in (G_1' \bigcap G_2')g \tag{3.6}$$

where $g \in (G_1' \bigcap G_2'f')f''$.

When the intersection is null, it implies that the specified spatial relationship is impossible, i.e. no positions for l_1, l_2 can realize the required contacts. When the intersection is not null, the relative position can be obtained by calculating a group intersection and choosing a particular element g.

When the spatial relationship is realizable, the two primitive features of B_1 can be viewed as one compound feature $F_1 = F_{11} \cup F_{12}$ fitting with another compound feature $F_2 = F_{21} \cup F_{22}$ of B_2. The common symmetry group of these two compound features can be obtained from $sym(F_{11}), sym(F_{12})$ or $sym(F_{21}), sym(F_{22})$. Following (3.2) we have:

$$l_1^{-1}l_2 \in f_1 sym(F_1) f_2 \tag{3.7}$$

where f_1, f_2 are the locations of the compound features F_1, F_2 with respect to their body coordinate systems. When the primitive features are distinct from each other, by Proposition 3.18,

$$l_1^{-1}l_2 \in f_1(sym(F_{11}) \cap sym(F_{12}))f_2. \tag{3.8}$$

In the case where $sym(F_{11}) \cap sym(F_{12})$ is the identity group, $\{1\}$, we have

$$l_1^{-1}l_2 = f_1 f_2^{-1}, \tag{3.9}$$

and the relative position of B_1 to B_2 is uniquely determined. Interestingly, the most asymmetrical case appears in the simplest form under this formulation.

Fitting relationships are quite common in assembly and are of primary concern in this thesis. Table 3.3 exhibits all the kinematic joints that are formed by fitting (lower pairs as shown in Figure 1.2) and the associated symmetry groups of the contacting features.

3.5 Discussions on Oriented Features

Whether a feature has orientations or not, or which orientation it has, does not make a difference in regards to the symmetries of the feature. For example, a

Table 3.3: Continuous Group and Degree of Freedom

Dimension (d.o.f.)	Symmetry Group (constraint)	Associated Lower pair
1	T^1	*Prismatic*
1	$SO2$	*Revolute*
1	$\mathcal{G}_{screw}$	*Screw*
2	$\mathcal{G}_{cylinder}$	*Cylindrical*
3	$\mathcal{G}_{plane}$	*Planar*
3	$SO3$	*Spherical*

spherical surface treated as a set, or with orientation vectors pointing inward, has the same symmetries as the spherical surface with orientation vectors pointing outward. This is why the treatment of primitive features as sets is sufficient as far as their symmetries are concerned. The only exception is the plane surface: when it is treated as a set there are flipping symmetries which do not exist for oriented planes. In order to characterise and distinguish features precisely, it is useful to characterize the orientation of a feature explicitly. Let $S2$ be the unit sphere at the origin, each point of which corresponds to a unit vector in $\Re^3$. Thus Definition 3.11 for primitive features can be extended as follows:

Definition 3.23 *An* **oriented primitive feature** $F = (S, \rho)$ *is an oriented surface where*

1) $S \subset \Re^3$ *is a connected, irreducible and non-trivial algebraic surface in Euclidean space*

2) $\rho \subset S \times S2$ *is a relation such that for all* $s \in S$, $(s, v) \in \rho$ *where* v *is one of the normal vectors of surface* S *at point* s.

Intuitively speaking, a feature is composed of both "skin", S , and "hair", the set of normal vectors which correspond to the points on $S2$.

We now define how an isometry acts on the relation ρ:

Definition 3.24 *Any isometry* $g = tr$ *of* $\mathcal{E}^+, t \in \mathbf{T}^3, r \in SO(3)$ *acts on* ρ *in such a way that*

$$(gs, rv) \in g * \rho \Leftrightarrow (s, v) \in \rho$$

where $s \in S, v \in S2$.

Next we prove the associativity of isometries when they act on the relation ρ.

Lemma 3.25 *For all* $g_1, g_2 \in \mathcal{E}^+, (g_1 g_2) * \rho = g_1 * (g_2 * \rho)$

Proof:

Let $g_1 = t_1 r_2, g_2 = t_2 r_2$ where $t_1, t_2 \in \mathbf{T}^3, r_1, r_2 \in SO(3)$ (For a justification see Section 4.1). Since $g_1 g_2 = t_1 r_1 t_2 r_2 = t_1 t' r_1 r_2$ ($\mathbf{T}^3$ is a normal subgroup of $\mathcal{E}^+$), for all $(s, v) \in \rho, (g_1 g_2 s, r_1 r_2 v) \in (g_1 g_2) * \rho$. On the other hand, for all $(s, v) \in \rho, (g_2 s, r_2 v) \in g_2 * \rho$ and $(g_1 g_2 s, r_1 r_2 v) \in g_1 * (g_2 * \rho)$. Therefore, $(g_1 g_2) * \rho = g_1 * (g_2 * \rho)$. □

For a feature defined in Definition 3.23, its symmetries are different from the symmetries of a set:

Definition 3.26 *An isometry* g *is a* **symmetry of a feature** $F = (S, \rho)$ *if and only if* $g(S) = S$ **and** $g * \rho = \rho$.

There is, therefore, an extra demand on the symmetries for an oriented feature, namely, these isometries not only keep the feature setwise invariant but also preserve its orientations. Since orientations are points on $S2$, such symmetries keep **two** sets of points setwise invariant *simultaneously*.

Proposition 3.27 *The symmetries of a feature* F *form a group, called the* **symmetry group of feature** F.

Proof:

Let G denote the set of the symmetries of F. Since it has shown in Proposition 3.13 that the proposition is true for set S, here we only state about ρ.

Obviously, $1 * \rho = \rho$, so $1 \in G$. If $g \in G$ then $(g * \rho) = \rho$. Multiplying by g^{-1} we have $g^{-1}(g * \rho) = g^{-1} * \rho$. Using Lemma 3.25 we have $g^{-1} * \rho = \rho$ and so $g^{-1} \in G$. Finally, if $g_1, g_2 \in G$ then $(g_1 g_2) * \rho = g_1 * (g_2 * \rho) = g_1 * \rho = \rho$ therefore $g_1 g_2 \in G$. Hence G is a group. $\mathcal{E}^+$. □

Now we need to redefine our definition for distinct features, taking orientations into consideration.

Definition 3.28 *Two primitive features* $F_1 = (S_1, \rho_1), F_2 = (S_2, \rho_2)$ *are said to be* **distinct** *if for all the open subsets* $S_1' \subset S_1, S_2' \subset S_2$*, and sub-relations* $\rho_1' = \{(s_1, v_1) \in \rho_1 | s_1 \in S_1'\}, \rho_2' = \{(s_2, v_2) \in \rho_2 | s_2 \in S_2'\}$*, no* $g = tr \in \mathcal{E}^+$ *exists such that* $g(S_1') = S_2'$ *and* $g * \rho_1' = \rho_2'$.

This definition allows us to distinguish a feature from its complement. Complement features are distinct from each other, except for the planar surfaces which are 2-congruent of each other. Nevertheless, complement features have the same symmetry group.

Definition 3.29 *A* **compound feature** $F = (S, \rho)$ *of primitive features* $F_1, \ldots, F_n$, *is defined to be*

- $S = S_1 \cup \ldots \cup S_n$
- $\rho = \rho_1 \cup \ldots \cup \rho_n$

A relation ρ instead of a function is chosen to denote the orientation of a feature because, when two primitive features are combined, there may be two distinct normal directions at one point of the feature, such as the edge where two planes meet.

Although Definition 3.11 for features is sufficient to capture the symmetry property when bodies are in contact, Definition 3.23 for the oriented primitive feature followed by Definition 3.29 for oriented compound feature is a more precise formalization that we intent to pursue further.

3.6 Summary

In this Chapter we have investigated how group theory, especially as regards the Euclidean group, is related to assembly. We formally defined:

- primitive and compound set features
- primitive and compound oriented features
- symmetry groups for features
- three comparison standards for features, namely, distinct, 1-congruent and 2-congruent.
- spatial relationship τ between two bodies

We have proved:

- The symmetries of any subset of the Euclidean space form a group.
- The symmetries of an oriented feature form a group.

- The symmetry group of a compound feature is the intersection of the symmetry groups of the component primitive features, when the primitive features are
  - distinct; or
  - separated and 1-congruent;
- The symmetry group of a compound feature is the intersection of the symmetry groups of the component primitive features multiplied by the smallest group generated by g_c i.e. $< g_c > (G_1 \cap G_2)$, when the the primitive features are separated and 2-congruent via g_c.

Two main points in this chapter are:

1) how the symmetry groups of features are being used to characterize features, primitive or compound; and

2) how the relative positions of two bodies in an assembly can be determined by the symmetry groups of the contacting features while they fit.

In particular, we examined the spatial relationship *fitting*, i.e. areal contact between bodies in an assembly. One basic observation is that when two bodies fit, they share the same surface at the area of contact and thus share the same symmetry group. This is of essential importance in understanding how features mate and what the final assembly configurations can be.

An extension to the definition of primitive features is given, which specifies the orientation(s) of a feature explicitly. It is a more precise and general formalization for features. Further exploration is worthwhile.

What we have described in this chapter provides a framework for determining the symmetry group of compound features and the relative positions of two bodies fitted through multiple features. Both of these require the computation of group intersection and, inevitably, a compact representation of a symmetry group which group can be finite or infinite, discrete or continuous. These will form the focus of the next chapter.

CHAPTER 4

TR SUBGROUPS OF THE PROPER EUCLIDEAN GROUP

The hidden and the manifest give birth to each other.
Difficult and easy complement each other.
Long and short exhibit each other.
High and low set measure to each other.
Voice and sound harmonize each other.
Back and front follow each other.

Lao Tzu, Tao Teh Ching

Assembly establishes spatial relationships between body features. It has been shown in earlier work [33, 63], that group theory, the mathematical treatment of symmetry, provides a powerful way of characterising many important spatial relationships. This approach associates the symmetry group as a major attribute of body features [48, 67]. All symmetry groups considered in this work are subgroups of the proper Euclidean group, $\mathcal{E}^+$, which consists of all proper isometries, i.e. mappings from $\Re^3$ to $\Re^3$ that preserve distance and handedness. When two bodies relate through multiple *fitting* features, it is possible to characterise their relative positions in terms of the intersection of the symmetry groups of the individual features. Hence, the goal of this chapter is to provide a representation[1] for these symmetry groups which lends itself to the implementation of an efficient group intersection algorithm.

It is possible to define a similarity metric between two isometries (Chapter 3). Under this metric $\mathcal{E}^+$ is a connected topological space. Subgroups of $\mathcal{E}^+$ that consist entirely of isolated points in the topology are said to be discrete. The discrete symmetry groups may be either finite or countably infinite, and consist of:

- the *dihedral groups*, D_{2n}, which are the symmetries of regular prisms;

[1]When we use the term *representation* as applied to groups, we mean computational structures which serve to *denote* groups, and *not* a representation in terms of groups of matrices, as is understood by group theorists.

- the *cyclic groups*, C_n, which are the symmetries of pyramids whose base is a regular polygon (excepting the regular tetrahedron);
- the groups of the Platonic solids, and
- the crystallographic groups, which are the symmetries of spatial lattices of regularly repeated geometric patterns.

All non-discrete subgroups of $\mathcal{E}^+$ contain points which have other points of the subgroup arbitrarily close to them. They include, for example, the symmetry group of the sphere, $SO(3)$, and the symmetry group of the plane, $\mathcal{G}_{plane}$.

The fact that both discrete and continuous topological structures exist within one group presents challenging problems for computation. Pure algebraic manipulation using computers is usually very expensive, thus does not provide the computational tractability one desires. This motivates us to search for alternatives.

Using a geometric representation to denote groups and compute group intersections is the theme of this chapter. Earlier work [68] developed a geometric treatment based on the idea of the *characteristic invariants* of a group. However, certain arbitrariness in the choice of characteristic invariants led to difficulties in extending the work beyond a limited class of groups. In this chapter, I provide a definition of characteristic invariants of a group which establishes a one-to-one correspondence between groups and their invariants. This permits a systematic treatment of an important class of subgroups of $\mathcal{E}^+$, namely the **TR** groups, each of which is a semidirect product of a translation subgroup T and a rotation subgroup R of $\mathcal{E}^+$. Further, I prove that the intersection of **TR** groups can be easily obtained by computing their characteristic invariants geometrically.

To outline this chapter:

1) We review preliminaries on group actions and the Euclidean group (Section 4.1).

2) We define translational invariants for translation subgroups and rotational invariants for rotation subgroups, prove their one-to-one correspondences respectively, and prove that the intersection of groups can be obtained by computing their invariants (Sections 4.2, 4.3).

3) We define **TR** groups and their characteristic invariants, prove that there is a one-to-one mapping between the **TR** subgroups and their invariants, and prove

that the intersection of **TR** groups can be obtained from their characteristic invariants and that **TR** groups are closed under intersection (Section 4.4)

4) Finally, a description of an efficient **TR** subgroup intersection algorithm closes this chapter.

The following notational conventions are used in this chapter:

- We use the right associative function composition law $fg(x) = f(g(x))$.
- $\cong$ means *isomorphic*.
- $A \subset B$ means set A is a proper subset of B.
- $A \subseteq B$ means either $A \subset B$ or $A = B$.
- $\Leftrightarrow$ means *if and only if*, also *iff*
- $\Rightarrow$ denotes *implication*.
- $\cup$ means set union.
- s_0 denotes the origin of the world coordinate system.
- S_0 represents the unit sphere centered at the origin
- If r is a rotation then $\bar{r}$ denotes the axis of the rotation and θ_r denotes the angle of the rotation.
- $x \vee y$ is the *join* (least upper bound) of two elements x, y of a poset. When $\vee$ applies to two groups H and K, $H \vee K$ is the least group containing both H and K.
- $\bigwedge$ denotes the greatest lower bound of a set.
- $Aut(X)$ is the automorphism group of X. When X is a set, $Aut(X)$ contains all the bijections from X to X.
- ∞ denotes the cardinal infinity corresponding to $|\Re|$ i.e. the order of the set reals.
- $\mathcal{N}$ denotes the set of natural numbers.

Definitions of algebraic terms used in this chapter can be found in Appendix A.

4.1 Preliminaries

Those definitions and facts that can be found in basic group theory texts [17, 24, 27, 55, 58] will be assumed as given. However, in this section we shall state the group theoretic definitions and propositions that we need, and prove those which are not easily found in text books.

4.1.1 Abstract Groups

The product of two groups H and K can be generalized to the construction of a **semidirect** product $H \times_\theta K$ depending on H, K and on a morphism $\theta : K \to Aut(H)$. We write $\theta(k) : H \to H$ as $k * h$ where $h \in H, k \in K$.

Definition 4.1 Semidirect Product of groups H and K ([55] p. 414)
For all $h_1, h_2 \in H$ and all $k_1, k_2 \in K$, the identities $k_1 * (h_1 h_2) = (k_1 * h_1)(k_1 * h_2)$, and $(k_1 k_2) * h_1 = k_1 * (k_2 * h_1)$ hold. For all the pairs of elements $(h_i, k_i) \in H \times_\theta K$ we define a product by

$$(h_1, k_1)(h_2, k_2) = (h_1(k_1 * h_2), k_1 k_2)$$

$H \times_\theta K$ is called the **semidirect product** *of H and K, relative to θ.*

When $\theta : K \to Aut(H)$ is the trivial morphism which takes every $k \in K$ to the identity automorphism 1_H, the semidirect product $H \times_\theta K$ reduces to the direct product $H \times K$.

Definition 4.2 *If a subgroup H of G for which $gHg^{-1} \subseteq H$ holds for all $g \in G$, then H is called a* **normal subgroup** *of G, written as $H \triangleleft G$.*

Proposition 4.3 *Any group G with subgroups $H \triangleleft G$ and K such that both $H \cap K = \{1\}$ and $H \vee K = G$ is a semidirect product. Explicitly, if $\theta : K \to Aut(H)$ assigns to each $k \in K$ the automorphism $h \mapsto khk^{-1}$ of H, an isomorphism $\Phi : H \times_\theta K \cong G$ is given by $\Phi(h, k) = hk$. ([55] p. 415)*

4.1.2 Group Actions

Definition 4.4 *A group G* **acts on** *a set X if there exists a mapping $\phi : G \times X \to X$, denoted by $(g, x) \mapsto g(x)$, which satisfies: $g(h(x)) = (gh)(x)$ and $e(x) = x$ where $g, h, e \in G, x \in X$.*

By this definition, one necessary condition for G to act on X is $G(X) = \{g(x)|x \in X, g \in G\} \subseteq X$.

Definition 4.5 *For each $x \in X$, $G(x) = \{g(x)|g \in G\}$ is called the* **orbit** *of x* **under the group** G.

If G acts on X then G may or may not act on a proper subset of X. For example, consider X as the set of all the vertices on a cube and Y be the set of vertices on one of the faces of the cube. The symmetry group G of the cube acts on X. Actually, $G(X) = X$. However, G cannot act on any proper subset Y of X, simply because some $g \in G$ brings some $y \in Y$ out of Y. Nevertheless, a subgroup of G, a cyclic group of order 4, for example, can act on both X and Y (Figure 4.1). Note that when C_4 acts on Y there is only one orbit, while C_4 acting on X produces two disjoint orbits.

Definition 4.6 *If G acts on X and if for each pair $x, y \in X$, there exists at least one $g \in G$ such that $y = g(x)$, then* **group G acts transitively on set X** *i.e. the orbit of every element of X under group G is X itself.*

In Figure 4.1 the symmetry group G of the cube acts on both X and Y transitively although C_4 acts on Y transitively, it does not act transitively on X.

Definition 4.7 *For all $x \in X$ the set $G^x = \{g \in G|gx = x\}$ is a subgroup of G called the* **stabilizer subgroup (isotropy group)** *of G at x.*

G^x contains those elements of G which leave x invariant.

Definition 4.8 *If group G acts on X then $\mathcal{F}_G = \{x \in X|\forall g \in G, g(x) = x\}$ is said to be the* **fixed-point set** *of G.*

Every point in the fixed-point set of G has G as its stabilizer group.

In the following two propositions let G_1, G_2 be subgroups of G, and let G_1, G_2 be conjugate, i.e. $G_1 = gG_2g^{-1}$ for some $g \in G$.

Proposition 4.9 $\mathcal{F}_{G_1} = g(\mathcal{F}_{G_2})$.

Proof:

Since $G_1 = gG_2g^{-1}$, for all $g_1 \in G_1$ there exists a $g_2 \in G_2$ such that $g_1 = gg_2g^{-1}$. For all $x \in \mathcal{F}_{G_2}, g_1(g(x)) = gg_2g^{-1}(g(x)) = gg_2(x) = g(x)$. Therefore $g(x) \in \mathcal{F}_{G_1}$ and so $g(\mathcal{F}_{G_2}) \subseteq \mathcal{F}_{G_1}$.

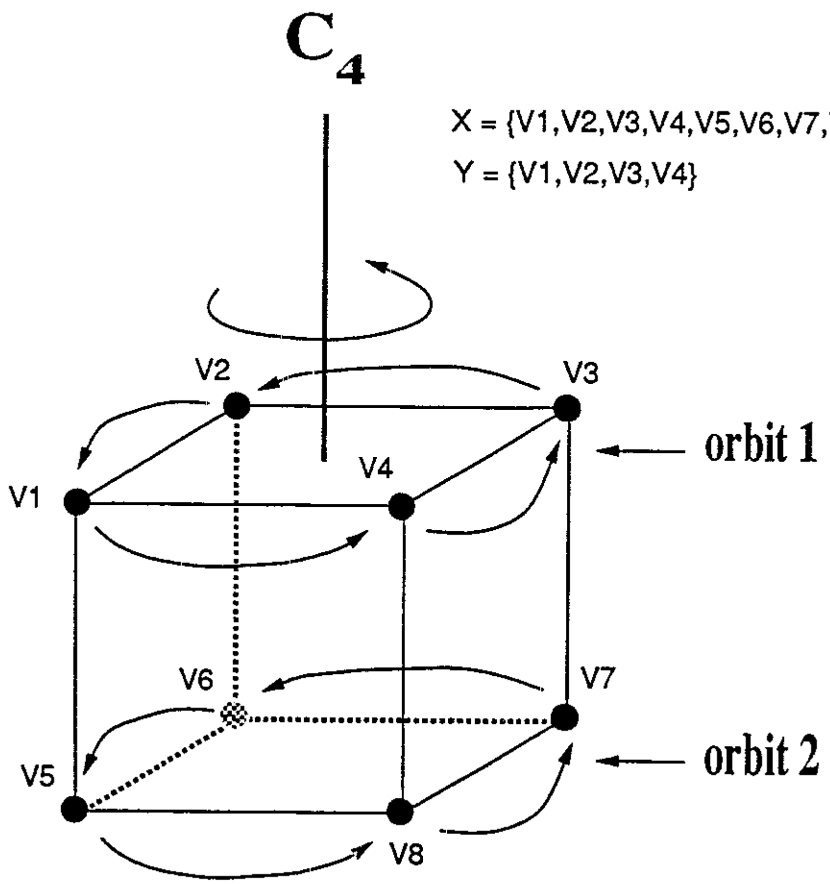

Figure 4.1: C_4 does not act on X transitively — there are two disjoint orbits; C_4 acts on Y transitively — there is only a single orbit.

Similarly, since for all $g_2 \in G_2$ there exists a $g_1 \in G_1$ such that $g_2 = g^{-1} g_1 g$, we have $\mathcal{F}_{G_1} \subseteq g(\mathcal{F}_{G_2})$.

Thus we conclude $\mathcal{F}_{G_1} = g(\mathcal{F}_{G_2})$. □

Proposition 4.10 $G_1^{g(p)} = g G_2^p g^{-1}$.

Proof:

For all $g_1 \in G_1$ there exists $g_2 \in G_2$ such that $g_1 = g g_2 g^{-1}$. In particular, if $g_1 \in G_1^{g(p)}$ then $g_1(g(p)) = g g_2 g^{-1}(g(p)) = g g_2(p) = g(p)$. Thus $g_2(p) = p$, so $g_2 \in G_2^p$ and $g_1 \in g G_2^p g^{-1}$. Therefore $G_1^{g(p)} \subseteq g G_2^p g^{-1}$.

Similarly, since for all $g_2 \in G_2^p$ there exists a $g \in G_1$ such that $g_2 = g^{-1} g_1 g$, $g G_2^p g^{-1} \subseteq G_1^{g(p)}$.

So we conclude $G_1^{g(p)} = g G_2 g^{-1}$. □

Proposition 4.11 *If* $G = G_1 \cap G_2$ *then* $G^p = G_1^p \cap G_2^p$.

Proof:

If $g \in G_1^p \cap G_2^p$ then $g(p) = p \Rightarrow g \in G^p$. Thus $G_1^p \cap G_2^p \subseteq G^p$.

If $g \in G^p$ then $g(p) = p$. Thus $G = G_1 \cap G_2$, $g \in G_1, g \in G_2$. Precisely, $g \in G_1^p, g \in G_2^p \Rightarrow g \in G_1^p \cap G_2^p$. Then $G^p \subseteq G_1^p \cap G_2^p$.

Hence $G^p = G_1^p \cap G_2^p$. □

4.1.3 The Euclidean Group and its Subgroups

Let us recall the definitions for the Euclidean group and the proper Euclidean group given in Section 3.1:

> The set of all the isometries of $\Re^3$ forms a group with the composition of isometries as group product; this group is called the **Euclidean group**. If we restrict the set of isometries to those isometries that preserve handedness, thus excluding reflections, then the remaining set still has a group structure under isometry composition called the **Proper Euclidean group**, denoted by $\mathcal{E}^+$.

In this section we shall explore the internal structure of the proper Euclidean group $\mathcal{E}^+$ using its subgroups as building blocks. Let us start by defining the distance of two sets in $\Re^3$ using the definition for the distance of two points in $\Re^3$ **dist** (Definition 3.7):

Definition 4.12 *If S_1 and S_2 are non-empty subsets of $\Re^3$, then the distance of sets S_1, S_2,* $\mathbf{dist}(S_1, S_2)$, *is defined to be*

$$\mathbf{dist}(S_1, S_2) = \bigwedge_{s_1 \in S_1 s_2 \in S_2} \mathbf{dist}(s_1, s_2).$$

Note, it may happen that $\mathbf{dist}(S_1, S_2) \neq \mathbf{dist}(s_1, s_2)$ for any $s_1 \in S_1, s_2 \in S_2$

Lemma 4.13 *If* $\mathbf{dist}(S_1, S_2) = \mathbf{dist}(s_1, s_2)$ *then for all subsets H_1, H_2 such that $s_1 \in H_1 \subseteq S_1, s_2 \in H_2 \subseteq S_2$,* $\mathbf{dist}(H_1, H_2) = \mathbf{dist}(S_1, S_2)$.

In Chapter 3 isometries were introduced as **distance preserving** mappings in $\Re^3$. Let us now consider all isometries that are **linear**. It has been proved [69] that all the linear isometries O are **length preserving,** i.e. for all $x \in \Re^3, \|O(x)\| = \|x\|$. Furthermore, all linear, length preserving mappings are orthogonal and form a group. This is how the following group gets its name.

Definition 4.14 *$O(3)$ is defined to be the* **real orthogonal group** *in three-space that contains all the linear isometries in $\Re^3$.*

We identify linear isometries in $\Re^3$ with their matrix representation [4, 24, 69] $O(3) = \{3 \times 3 \text{ matrices } O | OO^t = I_3\}$. Here I_3 is the 3×3 identity matrix, and for all $O \in O(3), \det(O) = \pm 1$.

$O(3)$ is the symmetry group of S_0. It contains rotations as well as reflections. To exclude reflections, we must constrain the determinant:

Definition 4.15 The special orthogonal group *$SO(3)$ is defined to be $SO(3) = \{O \in O(3) | \det(O) = +1\}$.*

Definition 4.16 *Any subgroup of $SO(3)$ is called a* **canonical rotation subgroup**[2] *of $\mathcal{E}^+$ and will be denoted by R_c.*

Definition 4.17 *R is a* **rotation subgroup** *of $\mathcal{E}^+$ iff R belongs to the conjugation class of a canonical rotation group, i.e. there exist a canonical rotation subgroup R_c and some $g \in \mathcal{E}^+$ such that $R = gR_cg^{-1}$.*

Definition 4.18 *Suppose R is a rotation group. If $\mathcal{F}_R$ is a single point then R is called a* **point group***; if $\mathcal{F}_R$ is a line then R is called a* **line group**.

[2] Do not confuse *canonical symmetry groups* (Chapter 3) with the *canonical rotation subgroups* which are all the subgroups of $SO(3)$.

Note, this notation differs from that used in Crystallography and other fields where R is a point group if $\mathcal{F}_R$ is not empty. Here, each member of a rotation group R keeps all the points of its fixed point set $\mathcal{F}_R$ *pointwise* fixed.

Proposition 4.19 *The elements of $SO(3)$ are rotations about s_0, i.e. for all $r \in SO(3), s_0 \in \bar{r}$. ([58] Theorem 2.1)*

Corollary 4.20 *A nontrivial canonical rotation group R_c is either a line group or a point group.*

Corollary 4.21 *A nontrivial rotation group R is either a line group or a point group.*

Proof:

By Definition 4.17, $R = tR_ct^{-1}$. By Proposition 4.9, $\mathcal{F}_R = t(\mathcal{F}_{R_c})$. By Corollary 4.20, $\mathcal{F}_{R_c}$ is either a point or a line passing through the origin. Thus $\mathcal{F}_R$ is either a point or a line translated by t from the origin. □

Note: Each rotation subgroup R treated in this paper belongs to one of the following two categories:

- If R has infinite cardinality, it is conjugate to $SO(3)$, $O(2)$ or $SO(2)$[3].
- If R has a finite order, it is conjugate to a cyclic group C_n, a dihedral group D_{2n} or one of the platonic groups – the symmetry groups for regular polyhedrons.

Definition 4.22 *A* **discrete group** *G is a subgroup of $\mathcal{E}$ such that for any $x \in \Re^3$ and any sphere $B_r = \{y \in \Re^3 : ||y|| \leq r\}$ there are only a finite number of points in the G-orbit of x that are contained in B_r.*

Definition 4.23 *Let T^3 be all the translations in the Euclidean group $\mathcal{E}$. T^3 has a group structure and is called the* **translation group** *of $\mathcal{E}$.*

Informally, we identify each translation t in T^3 with a point x in $\Re^3$. A translation t acts on a point p of $\Re^3$ by moving the point along the vector (Figure 4.2). That is, T^3 is isomorphic to $\Re^3$ under addition and therefore has a group structure which is connected and abelian. So there exists a bijection $\eta : \mathrm{T}^3 \rightarrow \Re^3$ such that for all translations $t_x, t_p \in \mathrm{T}^3$, we have $x = \eta(t_x), p = \eta(t_p) \in \Re^3$, and

[3] We exclude those infinite groups that are dense in $SO(3)$, e.g. isomorphic to the rationals. However, the *topological closure* of any infinite subgroup of SO(3) is one of these 3 groups.

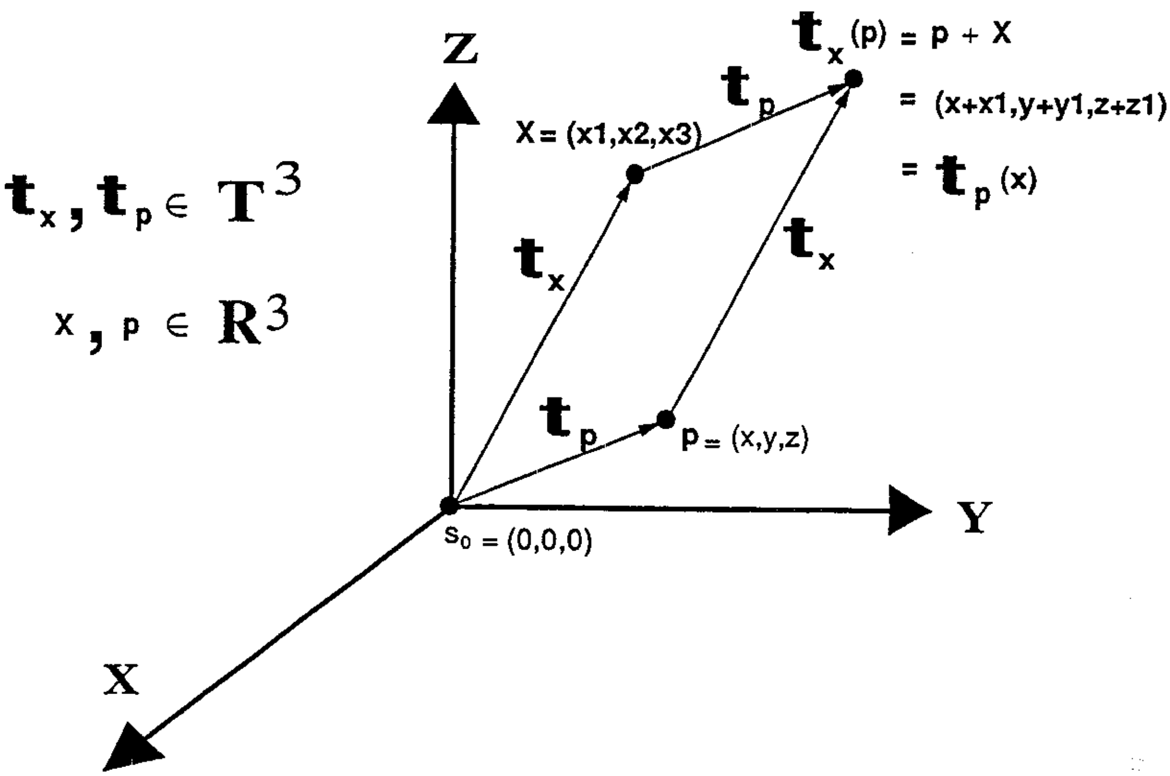

Figure 4.2: Identifying T^3 with $\Re^3$

$$\eta(t_x t_p) = \eta(t_x) + \eta(t_p) = x + p \qquad \textit{homomorphism property.}$$

In particular, a translation acting on a point of $\Re^3$ can be written as $t_x(p) = p + \eta(t_x) = p + x = \eta(t_p) + x = t_p(x)$ (Figure 4.2).

Each non-identity translation $t \in \mathrm{T}^3$ is a non-linear isometry, since for any pair of points $p_1, p_2 \in \Re^3, t_x(p_1+p_2) = p_1+p_2+x$, while $t_x(p_1)+t_x(p_2) = p_1+x+p_2+x = p_1+p_2+2x$. This non-linearity is the reason why translations can not be given a representation as 3×3 matrices.

An element of $\mathcal{E}^+$ can always be expressed as the product of a rotation and a translation, or can be seen as a screw-motion tr, where $t \in \mathrm{T}^3, r \in SO(3)$ and $t \| \bar{r}$, which includes the rotations and translations as 'extreme' cases [24]. **Note**, when $g = tr, t \perp \bar{r}$, g is a rotation [58].

Proposition 4.24 *If $g \in \mathcal{E}^+$ then $g = tr$ for a unique $t \in \mathrm{T}^3$ and a unique $r \in SO(3)$.*

Proof:

If $g = t_1 r_1 = t_2 r_2$ for some $t_1, t_2 \in \mathrm{T}^3, r_1, r_2 \in SO(3)$, then $t_2^{-1} t_1 = r_2 r_1^{-1}$. For $t_2^{-1} t_1$ to be a translation and to fix the origin s_0, it must be true that $t_2^{-1} t_1 = 1 \Rightarrow t_1 = t_2$. Thus $r_1 = r_2$. □

It has been proved that a member of the proper Euclidean group $\mathcal{E}^+$ can be represented by a 4×4 matrix [4] as:

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} & a \\ r_{21} & r_{22} & r_{23} & b \\ r_{31} & r_{32} & r_{33} & c \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where the upper left 3×3 sub-matrix specifies the rotational part of the transformation (an element of $SO(3)$), and the right hand column specifies a translation by (a, b, c). Group multiplication is then simply matrix multiplication. This representation has been widely accepted in robotics [62] and is usually referred to as the 4×4 *homogeneous transformation* matrix.

Since any rotation r can be viewed as a rotation about some Z axis after a proper change of basis, w.l.o.g, let r represent a rotation about the Z axis and $t = (a, b, c)$ be an arbitrary translation under the orthogonal coordinate system whose Z axis coincides with $\bar{r}$.

Proposition 4.25 *Let $t' = rtr^{-1}$. Then t' is a translation and $t' = (a\cos(\theta_r) - b\sin(\theta_r), a\sin(\theta_r) + b\cos(\theta_r), c)$.*

Proof :

Using the 4×4 homogeneous transformation matrix representation for each element of $\mathcal{E}^+$ we have:

$$t = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix}, r = \begin{pmatrix} \cos\theta_r & -sin\theta_r & 0 & 0 \\ \sin\theta_r & \cos\theta_r & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and

$$rtr^{-1} = \begin{pmatrix} 1 & 0 & 0 & a\cos\theta_r - b\sin\theta_r \\ 0 & 1 & 0 & a\sin\theta_r + b\cos\theta_r \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.1}$$

Recall that the result of rotating a vector $t = (a, b)$ on the x-y plane by angle θ is:

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a\cos\theta - b\sin\theta \\ a\sin\theta + b\cos\theta \end{pmatrix}$$

Thus matrix (4.1) shows that rtr^{-1} is a translation resulting from rotating t about $\bar{r}$ by θ_r.

□

Proposition 4.26 T^3 *is a normal subgroup of* $\mathcal{E}^+$.

Proof:

For all $g = t_g r_g \in \mathcal{E}^+$ (Proposition 4.24) and all $t \in \mathrm{T}^3, gtg^{-1} = t_g r_g t r_g^{-1} t_g^{-1} = r_g t r_g^{-1} \in \mathrm{T}^3$ (Proposition 4.25). Therefore $\mathrm{T}^3 \lhd \mathcal{E}^+$ (Proposition 4.2). □

Now, we are ready to describe the proper Euclidean group in terms of its subgroups T^3 and $SO(3)$.

Proposition 4.27 *The* **Proper Euclidean group** $\mathcal{E}^+ = \mathrm{T}^3 \times_\theta SO(3)$ *is a semi-direct product of* T^3 *and* $SO(3)$ *relative to* $\theta(r) : \mathrm{T}^3 \to \mathrm{T}^3$, *where for all* $r \in SO(3)$, *all* $t \in \mathrm{T}^3, \theta(r) : t \mapsto rtr^{-1}$. *The product rule of* $\mathcal{E}^+$ *follows:*

$$(t_1, r_1)(t_2, r_2) = (t_1(r_1 t_2 r_1^{-1}), r_1 r_2).$$

Proof:

This follows directly from Propositions 4.3, 4.24 and 4.26 □

Definition 4.28 *A subgroup of* $\mathcal{E}^+$ *is a* **translation subgroup** *iff it is a subgroup of* T^3.

Proposition 4.29 *For any $t_1, t_2 \in \mathbf{T}^3$ and any $x \in \Re^3$, if $t_1(x) = t_2(x)$ then $t_1 = t_2$.*

Proof:

Since $\eta : \mathbf{T}^3 \rightarrow \Re^3$ is a bijection, $t_1(x) = x + \eta(t_1) = t_2(x) = x + \eta(t_2) \Rightarrow \eta(t_1) = \eta(t_2) \Rightarrow t_1 = t_2$. □

Proposition 4.30 *Let T be a translation subgroup. If $t \in T^3, p \in \Re^3$ such that $t(T(p)) = T(p)$ then $t \in T$.*

Proof:

Since $p \in T(p)$, $T(p) \neq \emptyset$. For all $x \in \mathbf{T}(p), t(x) \in \mathbf{T}(p)$. Since T acts on $T(p)$ transitively (Definition 4.6), for the two elements $x, t(x)$ of $T(p)$ there exists $t' \in T$ such that $t'(x) = t(x)$. By Proposition 4.29 $t = t' \in T$. □

Proposition 4.31 *trt^{-1} is a rotation about axis $t(\bar{r})$ with angle of rotation θ_r.*

Proof:

Let us prove this using some basic geometric arguments. Figure 4.3 illustrates the effect of trt^{-1} acting on an arbitrary point $x \in \Re^3$. Here point B is a point on $t(\bar{r})$ that realizes $\mathbf{dist}(x, t(\bar{r})) = \mathbf{dist}(x, B)$, i.e. B is the perpendicular projection of point x on line $t(\bar{r})$. Point A is point B translated by t^{-1}, obviously, $A = t^{-1}(B) \in \bar{r}$. To prove that trt^{-1} is a rotation about axis $t(\bar{r})$ with rotation angle θ_r is to prove that the two triangles $\{A, t^{-1}(x), rt^{-1}(x)\}$ and $\{B, x, trt^{-1}(x)\}$ are congruent to each other. Due to the fact that t and t^{-1} form parallel, equal length line-segments as shown in Figure 4.3, one can easily verify that all the corresponding edges of the two triangles are of equal length. Hence they are congruent. □

That trt^{-1} is a rotation about a new parallel rotation axis $t(\bar{r})$ with angle of rotation θ_r is not obvious from the matrix product of trt^{-1},

$$trt^{-1} = \begin{pmatrix} \cos\theta_r & -\sin\theta_r & 0 & a - (a\cos\theta_r - b\sin\theta_r) \\ \sin\theta_r & \cos\theta_r & 0 & b - (a\sin\theta_r + b\cos\theta_r) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

This is because trt^{-1} is not a linear rotation (a rotation about the origin) any more.

Corollary 4.32 *If $\bar{r}_1 \| \bar{r}_2$ and $\theta_{r_1} = \theta_{r_2}$, with respect to the same reference vector, then for all $t \in T^3$, $t(\bar{r}_1) = \bar{r}_2 \Rightarrow tr_1t^{-1} = r_2$.*

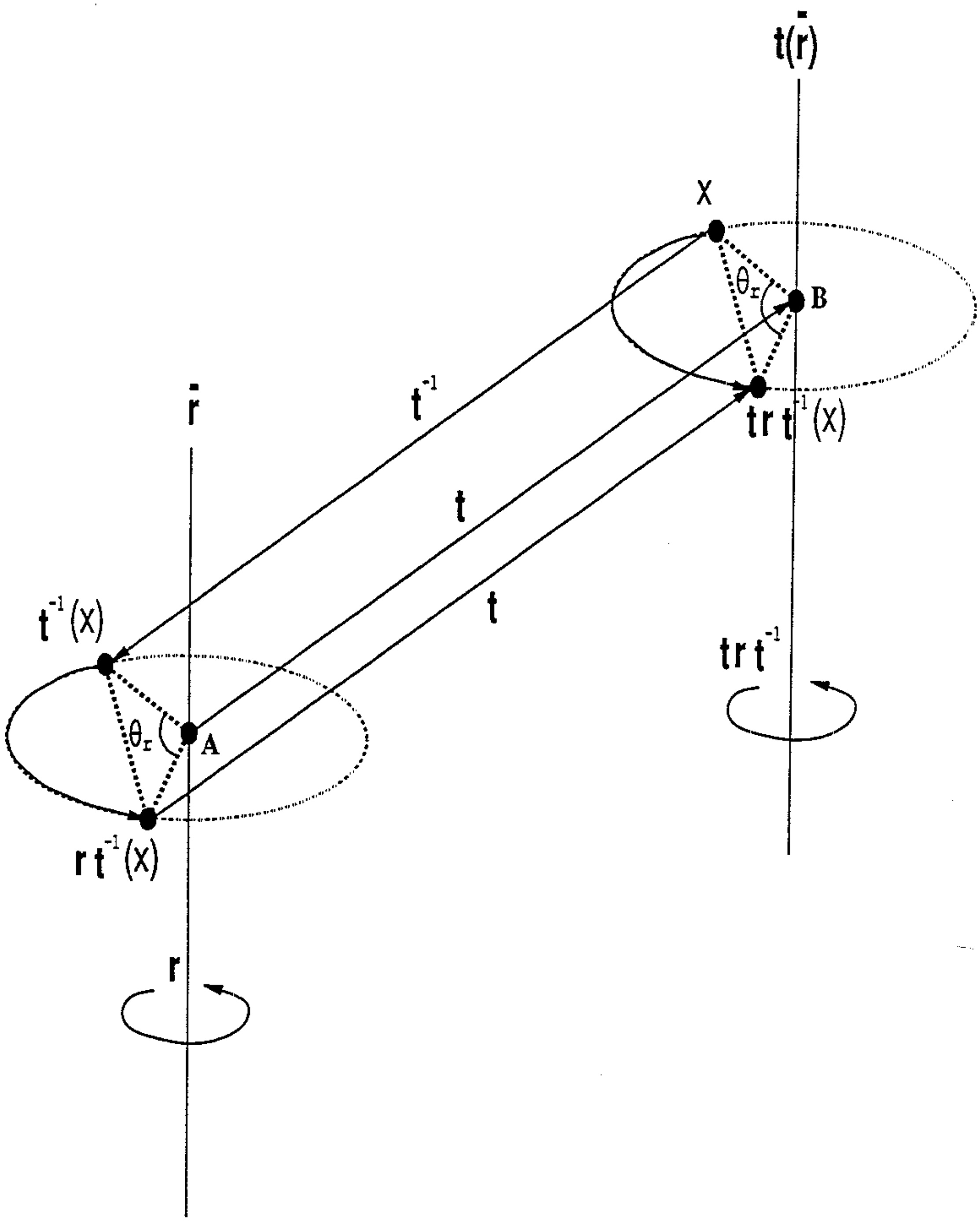

Figure 4.3: This shows the effect of trt^{-1} acting on an arbitrary point x in order to prove that trt^{-1} is a rotation about axis $t(\bar{r})$ with angle of rotation θ_r.

Proposition 4.33 *Let R_1, R_2 be rotation groups. If $R_1 = tR_2t^{-1}$ for some $t \in \mathbf{T}^3$ and $\mathcal{F}_{R_1} \cap \mathcal{F}_{R_2} \neq \emptyset$ then $R_1 = R_2$.*

Proof:

For all $r_1 \in R_1$ there exists $r_2 \in R_2$ such that $r_1 = tr_2t^{-1}$. By Proposition 4.31, $\bar{r}_1 \| \bar{r}_2$ and $\theta_{r_1} = \theta_{r_2}$. Since $\mathcal{F}_{R_1} \cap \mathcal{F}_{R_2} = \mathcal{F} \neq \emptyset$, there exists at least one point $x \in \mathcal{F}$ such that $x \in \bar{r}_1$ and $x \in \bar{r}_2$. Therefore $r_1 = r_2 \in R_2$ and so $R_1 \subseteq R_2$. Similarly, $R_2 \subseteq R_1$.

Thus we conclude $R_1 = R_2$.

Corollary 4.34 *If R_1, R_2 are canonical rotation groups and $R_1 = gR_2g^{-1}$ for some $g = tr$ with $t \in \mathbf{T}^3, r \in SO(3)$ then $R_1 = rR_2r^{-1}$.*

Proof: $R_1 = trR_2r^{-1}t^{-1}$. Since $rR_2r^{-1} \subseteq SO(3)$, $R_1 = rR_2r^{-1}$ (Proposition 4.33). □

Proposition 4.35 *If R is a rotation group then there exists a* **unique** *canonical rotation group R_c such that $R = tR_ct^{-1}$ for some $t \in \mathbf{T}^3$.*

Proof:

By Definition 4.17, R being a rotation subgroup means $R = gR_c'g^{-1}$ for some $g = tr \in \mathcal{E}^+$ where $t \in \mathbf{T}^3, r \in SO(3)$ and $R_c' \subseteq SO(3)$. Since $rR_c'r^{-1} = R_c \subseteq SO(3)$, we have $R = tR_ct^{-1}$.

Now assume there are two canonical rotation subgroups R_{c_1}, R_{c_2} and translations t_1 and t_2 such that $R = t_1R_{c_1}t^{-1} = t_2R_{c_2}t_2^{-1}$. Then $R_{c_1} = t_0R_{c_2}t_0^{-1}, t_0 = t_1^{-1}t_2 \in \mathbf{T}^3$. Since at least $s_0 \in \mathcal{F}_{R_{c_1}}, s_0 \in \mathcal{F}_{R_{c_2}}$, by Proposition 4.33 $R_{c_1} = R_{c_2} = R_c$. □

Definition 4.36 *Let R_1, R_2 be nontrivial rotation groups, $R_1 = tR_2t^{-1}$. Among all the $t_i \in \mathbf{T}^3$ such that $R_1 = t_iR_2t_i^{-1}$, let t_{min} be the* **minimum translation** *such that $||t_{min}|| = min\{||t_i||\}$.*

Proposition 4.37 *The minimum translation t_{min} is* **unique** *where $||t_{min}|| =$* **dist**$(\mathcal{F}_{R_1}, \mathcal{F}_{R_2})$.

Proof:

By Proposition 4.9, $R_1 = tR_2t^{-1} \Rightarrow \mathcal{F}_{R_1} = t(\mathcal{F}_{R_2})$. Thus $\mathcal{F}_{R_1}, \mathcal{F}_{R_2}$ are both points of $\Re^3$, or are two parallel lines (Figure 4.4).

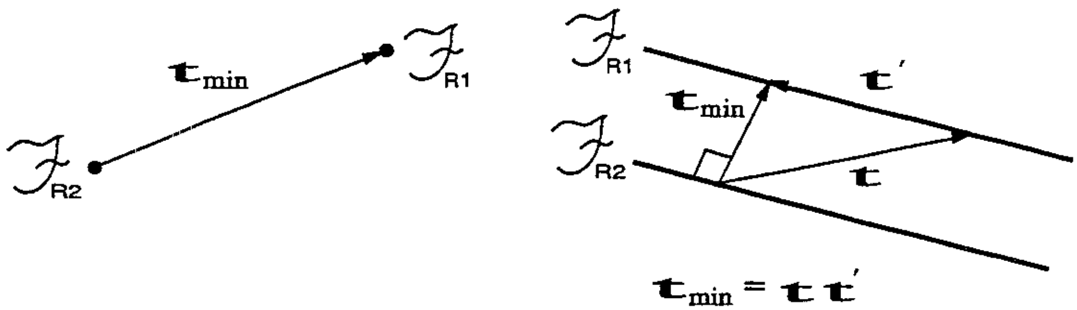

Figure 4.4: Minimum translation t_{min}

1. $\mathcal{F}_{R_1}, \mathcal{F}_{R_2}$ are two points in $\Re^3$. The only translation which moves one point to the other is $||t|| = \mathbf{dist}(\mathcal{F}_{R_1}, \mathcal{F}_{R_2}) = ||t_{min}||$ and so $t_{min} = t$ is the minimum translation as desired.

2. $\mathcal{F}_{R_1}, \mathcal{F}_{R_2}$ are parallel lines in $\Re^3$. Since $\mathcal{F}_{R_1}, \mathcal{F}_{R_2}$ are closed subspaces of $\Re^3$, $\mathbf{dist}(\mathcal{F}_{R_1}, \mathcal{F}_{R_2})$ can be identified with the distance of a pair of points from $\mathcal{F}_{R_1}, \mathcal{F}_{R_2}$ respectively. This determines the minimum translation between $\mathcal{F}_{R_1}$ and $\mathcal{F}_{R_2}$ to be the one that is perpendicular to both lines as shown in Figure 4.4. So $||t_{min}|| = \mathbf{dist}(\mathcal{F}_{R_1}, \mathcal{F}_{R_2})$.

   Now we need to prove that $R_1 = t_{min} R_2 t_{min}^{-1}$. Let t' be a translation along $\mathcal{F}_{R_1}$ such that $t_{min} = tt'$. From $R_1 = tR_2t^{-1}$, for all $r_1 \in R_1$ there exists $r_2 \in R_2$ such that $r_1 = tr_2t^{-1}$. Since $t' || \bar{r}_2, r_2 = t'r_2t'^{-1}$ (Proposition 4.31). Thus $r_1 = tt'r_2t'^{-1}t^{-1} = t_{min} r_2 t_{min}^{-1} \in t_{min} R_2 t_{min}^{-1}$. Hence $R_1 \subseteq t_{min} R_2 t_{min}^{-1}$. Similarly, we can prove $t_{min} R_2 t_{min}^{-1} \subseteq R_1$. Therefore $R_1 = t_{min} R_2 t_{min}^{-1}$. □

From now on, whenever we have $R_1 = tR_2t^{-1}$, t should be understood to be the unique minimum translation whose existence has just been proved.

Definition 4.38 *A* **pole** *of a rotation group $R = tR_ct^{-1}$ is a point p on S_0 which is left fixed by some rotation of the group R_c other than the identity. I.e for some $p \in S_0, \exists r \in R_c, r \neq 1$ such that $r(p) = p$.*

Each rotation in $SO(3)$ fixes a line passing through the origin. This axis of rotation pierces the unit sphere to form two poles, so poles always appear in pairs (Figure 4.5). Different rotations about the same axis have the same pair of poles. Therefore poles are not sufficient to be used to distinguish different rotation groups. Nevertheless, we shall see in the next proposition that the number of poles of a

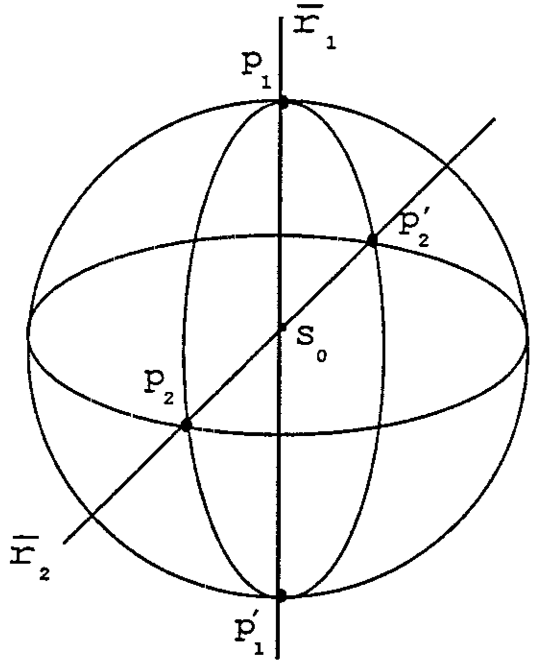

Figure 4.5: When a rotation group has only two poles p_1 and p_2, they have to belong to the same rotation axis; otherwise more than two poles would appear.

rotation group can determine whether a rotation group is a line or point group. Firstly,

Lemma 4.39 *$R_c = \cup R_c^p$ where R_c^p is the stabilizer subgroup of R_c at a pole p.*

Proof:

$R_c^p \subseteq R_c \Rightarrow \cup R_c^p \subseteq R_c$.

For each nontrivial rotation $r \in R_c$, there exists a $p \in S_0$ such that $r(p) = p$. By Definition 4.7 and Definition 4.38 there exists a stabilizer subgroup $R_c^p \subseteq R_c$ and $r \in R_c^p$. Therefore $r \in \cup R_c^p$. So $R_c \subseteq \cup R_c^p$ completes the proof. □

Corollary 4.40 *If $R = tR_ct^{-1}$, then $R = \cup R^{t(p)}$ for all the poles p of R.*

Proof: From Lemma 4.39 $R_c = \cup R_c^p \Rightarrow R = tR_ct^{-1} = t(\cup R_c^p)t^{-1} = \cup tR_c^p t^{-1} = \cup R^{t(p)}$. Here $R^{t(p)} = tR_c^p t^{-1}$ is a special case of Proposition 4.10. □

Proposition 4.41 *Let P be the set of poles of a canonical rotation group R_c. $|P| = 0$ iff $R_c = \{1\}$; $|P| = 2$ iff R_c is a line group, and $|P| > 2$ iff R_c is a point group.*

Proof:

Since $|P| = 0$, by Definition 4.38, for all $p \in S_0$ there exists no $r \neq 1$ such that $r(p) = p$. Then for all $p \in S_0, |R_c^p| = 1 \Leftrightarrow$ By Lemma 4.39 $R_c = \cup R_c^p = \{1\}$.

When $|P| = 2$, let $P = \{p_1, p_2\}$, $p_1 \neq p_2$. By the definition of poles, there exists at least one nontrivial rotation $r_1 \in R_c$ such that $r_1(p_1) = p_1$. Then $p_1, p_1' \in \bar{r}_1$ (Figure 4.5). If $p_2 \neq p_1'$ then $|P| = |\{p_1, p_2, p_1' \ldots\}| > 2$, a contradiction. Therefore $p_2 = p_1'$. Hence every $r \in R_c$ rotates about the line determined by p_1 and p_2, i.e. $R_c^{p_1} = R_c^{p_2} = R_c$ is a line group. Now, if R_c is a line group then $\mathcal{F}_{R_c}$ is a line passing through s_0. For all $r \in R_c, \bar{r} = \mathcal{F}_{R_c}$, since otherwise there would be a $r \in R_c, x \in \mathcal{F}_{R_c}$ such that $r(x) \neq x$, a contradiction of the definition of fixed point set. Therefore only two points on S_0 are left fixed by members of R_c and so $|P| = 2$.

When $|\mathcal{P}_R| > 2$, by Corollary 4.21 and the proof above, R cannot be a line group $\Leftrightarrow R$ is a point group. □

Obviously, this proposition is also true for any rotation subgroup of $\mathcal{E}^+$ that is not a canonical rotation group (Definition 4.17 and Proposition 4.9).

Proposition 4.42 *If R_1, R_2 are rotation line groups about same axis then $\gcd^*(|R_1|, |R_2|) = |R_1 \cap R_2|$ where*

$$\gcd^*(m,n) = \begin{cases} \gcd(m,n) & \text{if } m,n \text{ both are integers} \\ n & \text{if } m = \infty \\ m & \text{if } n = \infty \\ \infty & \text{if } a = b = \infty \end{cases}$$

Proof :

If one of R_1, R_2 has an infinite order, w.l.o.g. let $R_1 \cong SO(2)$. Clearly R_2 is a subgroup of R_1 and $R_1 \cap R_2 = R_2$. Thus $|R_1 \cap R_2| = |R_2| = \gcd^*(|R_1|, |R_2|)$.

If R_1, R_2 are both finite rotation groups, let $|R_1| = m, |R_2| = n, \gcd(m,n) = k$. There exists a cyclic group C_k such that $C_k \subseteq R_1 \cap R_2$ (p.63, theorem 5.3 [27]). Let $R_1 \cap R_2 = C_q$, a cyclic group of order q. By Lagrange's theorem, $k|q$. Thus there exists an integer c such that $q = ck$. Since also $q|m$ and $q|n$, we have $ck|m, ck|n$. However, k is the largest common divisor of m and n, so c must be 1. Therefore $q = k = \gcd(m,n)$. Hence $|R_1 \cap R_2| = \gcd^*(|R_1|, |R_2|)$. □

Corollary 4.43 $\gcd^*(|R_1|, |R_2|) = 1$ *iff* $R_1 \cap R_2 = \{1\}$

4.2 Translational Characteristic Invariants for Translation Subgroups

In this section we define translational invariants, which provide a geometric characterization of translation subgroups of $\mathcal{E}^+$.

Definition 4.44 *The* **translational invariant** $\mathcal{T}_T$ *of a group* $T \subseteq \mathbf{T}^3$ *is defined to be the T-orbit of the origin* s_0: $\mathcal{T}_T = T(s_0) = \{t(s_0) | t \in T\}$.

Clearly, $\forall t \in T, t(\mathcal{T}_T) = t(T(s_0)) = T(s_0) = \mathcal{T}_T$.

Definition 4.45 *If* $\mathcal{T}_T$ *is a manifold then* T *is called a* **continuous translation subgroup**. *Otherwise,* T *is called a* **discrete translation subgroup**, *where* $\mathcal{T}_T$ *can be a one, two or three dimensional lattice, a set of parallel lines, or a set of parallel planes in* $\Re^3$.

Let T_1, T_2 be translation subgroups of $\mathcal{E}^+$.

Proposition 4.46 *If* $\mathcal{T}_{T_1} = \mathcal{T}_{T_2}$ *then* $T_1 = T_2$.

Proof:

Let $\mathcal{T}_{T_1} = \mathcal{T}_{T_2} = \mathcal{T}$. For all $t_1 \in T_1$ there exists $x \in \mathcal{T}$ such that $x = t_1(s_0) \in \mathcal{T}$ (Definition 4.44 and Definition 4.6). There also exists $t_2 \in T_2$ such that $t_2(s_0) =$

$x = t_1(s_0)$. By Proposition 4.29 $t_1 = t_2 \in T_2$, thus $T_1 \subseteq T_2$. Similarly, $T_2 \subseteq T_1$. Therefore we conclude $T_1 = T_2$. □

We can thus define a *bijection* $\beta_t : \{T\} \to \{T(s_0)\}$ which maps all the translation subgroups to and only to their translational invariants.

Proposition 4.47 *The translational invariant of the intersection of T_1 and T_2 is the intersection of their translational invariants, i.e. $\mathcal{T}_{T_1 \cap T_2} = \mathcal{T}_{T_1} \cap \mathcal{T}_{T_2}$*

Proof:

If $x \in \mathcal{T}_{T_1 \cap T_2} = (T_1 \cap T_2)(s_0)$ then there exists $t \in T_1 \cap T_2$ such that $x = t(s_0) \in T_1(s_0) \cap T_2(s_0) = \mathcal{T}_{T_1} \cap \mathcal{T}_{T_2}$. Therefore $\mathcal{T}_{T_1 \cap T_2} \subseteq \mathcal{T}_{T_1} \cap \mathcal{T}_{T_2}$.

If $x \in \mathcal{T}_{T_1} \cap \mathcal{T}_{T_2}$ then $x \in T_1(s_0) \cap T_2(s_0)$. Thus $x = t_1(s_0) = t_2(s_0)$ where $t_1 \in T_1, t_2 \in T_2$. By Proposition 4.29, $t_1 = t_2 \in T_1 \cap T_2$. So $x \in (T_1 \cap T_2)(s_0) = \mathcal{T}_{T_1 \cap T_2}$. Therefore $\mathcal{T}_{T_1} \cap \mathcal{T}_{T_2} \subseteq \mathcal{T}_{T_1 \cap T_2}$.

So we conclude $\mathcal{T}_{T_1 \cap T_2} = \mathcal{T}_{T_1} \cap \mathcal{T}_{T_2}$. □

4.3 Rotational Characteristic Invariants for Rotation Subgroups

Now let us define the rotational invariants, which provide a geometric characterization of rotation subgroups of $\mathcal{E}^+$.

Let $R = tR_c t^{-1}$ be a rotation subgroup of $\mathcal{E}^+$, where $t \in \mathbf{T}^3, R_c \subseteq SO(3)$.

Definition 4.48 *The* **pole-set** $\mathcal{P}_R$ *of R is defined to be a set of pairs (p, n) where p is a pole and $n \in \mathcal{N} \cup \{\infty\}$, i.e.*

$$\mathcal{P}_R = \{(p, n) | p \in S_0, n = |R_c^p| > 1\}$$

where $|R_c^p|$ is the order of the stabilizer subgroup of R_c at p.

R_c^p is a *line group* since it fixes at least two distinct points p and s_0. Obviously, $\mathcal{P}_R = \mathcal{P}_{R_c}$. Hence all rotation subgroups R that are conjugate to each other via translations have the same pole-set $\mathcal{P}_R$, as is proved in the following:

Proposition 4.49 *Let $R_1 = t_1 R_{c_1} t_1^{-1}, R_2 = t_2 R_{c_2} t_2^{-1}$. $\mathcal{P}_{R_1} = \mathcal{P}_{R_2} \Leftrightarrow \exists t \in \mathrm{T}^3, R_1 = tR_2 t^{-1}$.*

Proof:

($\Rightarrow$) By Definition 4.48 $\mathcal{P}_{R_1} = \mathcal{P}_{R_{c_1}} = \mathcal{P}_{R_2} = \mathcal{P}_{R_{c_2}}$. For all $(p,n) \in \mathcal{P}_{R_{c_1}}, (p,n) \in \mathcal{P}_{R_{c_2}}$. That is to say that $R^p_{c_1} = R^p_{c_2}$ for all the poles p of R_{c_1}, R_{c_2}, since the stabilizer groups $R^p_{c_1}, R^p_{c_2}$ have the same axis of rotation and the same order n. Therefore $R_{c_1} = \cup R^p_{c_1} = \cup R^p_{c_2} = R_{c_2}$ (Lemma 4.39). Rewrite $R_{c_1} = R_{c_2}$ in terms of R_1, R_2 we have $t_1^{-1}R_1t_1 = t_2^{-1}R_2t_2 \Rightarrow R_1 = tR_2t^{-1}$ where $t = t_1t_2^{-1} \in \mathbf{T}^3$.

($\Leftarrow$) Since $R_1 = tR_2t^{-1} = tt_2R_{c_2}t_2^{-1}t^{-1}$, by Definition 4.48, $\mathcal{P}_{R_1} = \mathcal{P}_{R_{c_2}} = \mathcal{P}_{R_2}$.

□

Proposition 4.50 *Let R_1, R_2 be rotation subgroups. $R_1 = tR_2t^{-1}$ for some $t \in \mathbf{T}^3 \Leftrightarrow \mathcal{P}_{R_1} = \mathcal{P}_{R_2}$ and $\mathcal{F}_{R_1} = t\mathcal{F}_{R_2}$.*

Proof:

($\Rightarrow$) Given some $t \in \mathbf{T}^3, R_1 = tR_2t^{-1}$, by Proposition 4.9 $\mathcal{F}_{R_1} = t\mathcal{F}_{R_2}$, and from Proposition 4.49 $\mathcal{P}_{R_1} = \mathcal{P}_{R_2}$.

($\Leftarrow$) Since $\mathcal{P}_{R_1} = \mathcal{P}_{R_2}$, there exists a translation t' such that $R_1 = t'R_2t'^{-1}$ (Proposition 4.49). From Proposition 4.9 we have $\mathcal{F}_{R_1} = t'(\mathcal{F}_{R_2})$. For each $r_1 \in R_1$ there exists $r_2 \in R_2$ such that $r_1 = t'r_2t'^{-1} \Rightarrow \bar{r}_1 \| \bar{r}_2$ and $\theta_{r_1} = \theta_{r_2}$ (Proposition 4.31).

If R_1, R_2 are line groups, then $\mathcal{F}_{R_1} \| \mathcal{F}_{R_2}$. For all $r_1 \in R_1, r_2 \in R_2$, $t(\bar{r}_2) = \bar{r}_1$. By Corollary 4.32 $r_1 = tr_2t^{-1}$. Thus $R_1 \subseteq tR_2t^{-1}$. Similarly, $tR_2t^{-1} \subseteq R_1$. Therefore $R_1 = tR_2t^{-1}$.

If R_1, R_2 are point groups then $t = t'$ (Proposition 4.29). Therefore $R_1 = tR_2t^{-1}$

□

We see from this Proposition that both $\mathcal{P}_R$ and $\mathcal{F}_R$ are needed to characterise a rotation group. $\mathcal{P}_R$ tells the order of the rotations in R and $\mathcal{F}_R$ tells where in $\Re^3$ the rotation axes reside, so different rotation groups that are conjugate via a translation can be distinguished. We thus have the following definition:

Definition 4.51 *The* **rotational characteristic invariant of a rotation group** *R is the pair $(\mathcal{F}_R, \mathcal{P}_R)$, where $\mathcal{F}_R$ is the fixed point set of R (Definition 4.8) and $\mathcal{P}_R$ is the pole-set of R defined as above (Definition 4.48).*

Examples of rotational characteristic invariants of $SO(3)$, of the rotational symmetry group of a finite cylinder R_{cyl}, and of the symmetry group of a square-based pyramid C_4 are shown in Figure 4.6. For $SO(3)$, $\mathcal{P}_{SO(3)}$ is the whole of S_0. Thus each point on S_0 is a pole whose stabilizer group has an infinite order. The pole-set of R_{cyl} consists of the "north pole", the "south pole" and the "equator" of S_0, as

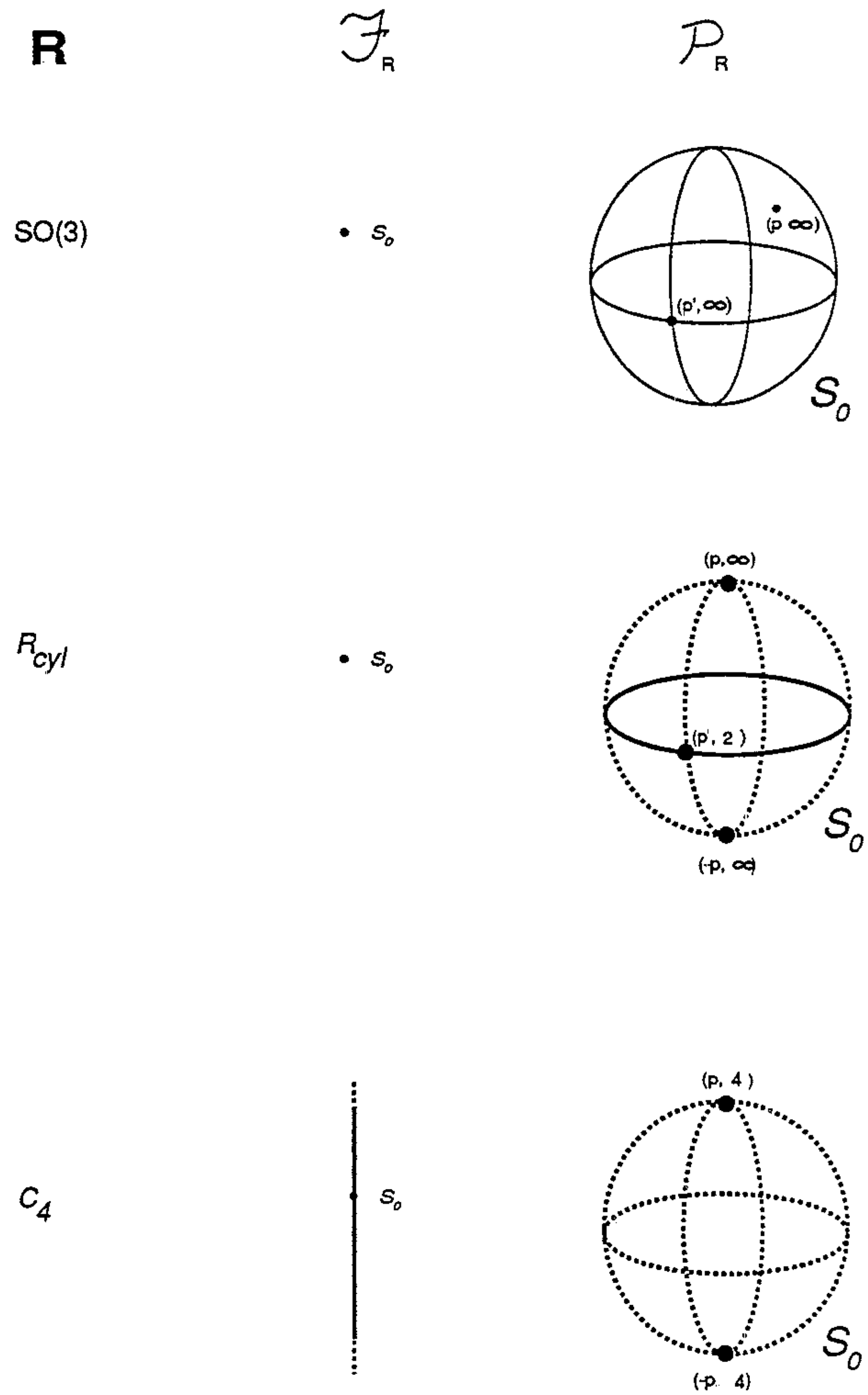

Figure 4.6: Examples for the fixed point set $\mathcal{F}_R$ and pole-set $\mathcal{P}_R$ of some rotation subgroups R.

shown in Figure 4.6. For C_4, only two points of S_0 are poles They come from the intersection of the rotation axis, which happens to be $\mathcal{F}_{C_4}$, and S_0. All the possible rotations of a rotation group can be obtained from this representation by re-centering the unit sphere S_0, which carries the poles of R, at any point of $\mathcal{F}_R$ (Proposition 4.9 and 4.10).

Since the stabilizer group R^p at a pole p is a line group, when $|R^p| = n$, there are $n-1$ nontrivial rotations about the axis determined by the pair of antipodal poles $(p, -p)$.

We now prove that $\mathcal{P}_R$ and $\mathcal{F}_R$ characterise the rotation group R completely.

Proposition 4.52 *Let R_1, R_2 be rotation groups. $\mathcal{F}_{R_1} = \mathcal{F}_{R_2}$ and $\mathcal{P}_{R_1} = \mathcal{P}_{R_2} \Rightarrow R_1 = R_2$.*

Proof:

This follows from Proposition 4.50. □

This proposition proves that there exists a bijection β_r which maps the set of rotation groups $\{R\}$ to the set of rotational invariants $\{(\mathcal{F}_R, \mathcal{P}_R)\}$, i.e. $\beta_r : \{R\} \to \{(\mathcal{F}_R, \mathcal{P}_R)\}$.

Given two rotation groups R_1 and R_2, we are looking for a computational method to construct both the pole-set of $R_1 \cap R_2$ i.e. to obtain $\mathcal{P}_{R_1 \cap R_2}$ from $\mathcal{P}_{R_1}$ and $\mathcal{P}_{R_2}$; and the fixed point set of $R_1 \cap R_2$. In the following Proposition we prove that the pole-set of the intersection of two rotation groups can be constructed by intersecting the pole-sets of each rotation group in a particular way. The reason why a standard set intersection of two pole-sets is inadequate can be illustrated by a counter example: the pole-sets of the cyclic groups C_6 and C_9 are: $\{((0,0,1),6),(0,0,-1),6)\}$, $\{((0,0,1),9),((0,0,-1),9)\}$, respectively. An intersection of the two pole-sets will result in an empty set. The correct pole-set that characterises this group intersection is $\{((0,0,1),3),((0,0,-1),3)\}$ This is reflected in $\gcd^*(|C_6|,|C_9|) = 3$, so the intersection should have the original poles but associated with order 3. For the cyclic groups C_3 and C_4 their pole-set are: $\{((0,0,1),3),((0,0,-1),3)\},\{((0,0,1),4),((0,0,-1),4)\}$. Since C_3 and C_4 as rotation groups do not have any common element other than the identity, the pole-set of their intersection should be the empty set. This is determined by the fact that $\gcd^*(|C_3|,|C_4|) = 1$ which corresponds to the identity group (Corollary 4.43). Therefore we define the following intersection operation $\cap^*$ on two pole-sets:

$\mathcal{P}_{R_1} \cap^* \mathcal{P}_{R_2} = \{(p,n) | \gcd^*(|R^p_{c_1}|,|R^p_{c_2}|) = n > 1\}$

Note, $\gcd^*(|R^p_{c_1}|,|R^p_{c_2}|) = n > 1$ implies $(p,|R^p_{c_1}|) \in \mathcal{P}_{R_1}$ and $(p,|R^p_{c_2}|) \in \mathcal{P}_{R_2}$.

Proposition 4.53 *Let $R = R_1 \cap R_2$ and $\mathcal{F} = \mathcal{F}_{R_1} \cap \mathcal{F}_{R_2}$ Then*

$$\mathcal{P}_R = \begin{cases} \mathcal{P}_{R_1} \cap^* \mathcal{P}_{R_2} & \text{if } \mathcal{F} \neq \emptyset \\ \mathcal{P}_{R_1} \cap^*_{\mathcal{L}_0} \mathcal{P}_{R_2} & \text{if } \mathcal{F} = \emptyset, \text{ and } R_1, R_2 \text{ are both point groups.} \\ \emptyset & \text{Otherwise} \end{cases}$$

and

$$\mathcal{F}_R = \begin{cases} \Re^3 & \text{if } |\mathcal{P}_R| = 0 \\ t_{\mathcal{F}}(\mathcal{L}_{p_1,p_2}) & \text{if } \mathcal{F} \neq \emptyset \text{ and } |\mathcal{P}_R| = 2 \\ \mathcal{F} & \text{if } \mathcal{F} \neq \emptyset \text{ and } |\mathcal{P}_R| > 2 \\ \mathcal{L} & \text{if } \mathcal{F} = \emptyset \text{ and } |\mathcal{P}_R| = 2 \end{cases}$$

where $\mathcal{P}_{R_i}$ is the pole-set of group R_i, $i = 1$ or 2;
$R_i^x \subseteq gSO(2)g^{-1}$ is the stabilizer *subgroup (a line group) of R_i at x;*
$\mathcal{P}_{R_1} \cap^* \mathcal{P}_{R_2} = \{(p,n) | \gcd^*(|R^p_{c_1}|, |R^p_{c_2}|) = n > 1\}$;
$\mathcal{P}_{R_1} \cap^*_{\mathcal{L}_0} \mathcal{P}_{R_2} = \{(p,n) | (p,n) \in \mathcal{P}_{R_1} \cap^* \mathcal{P}_{R_2}, p \in \mathcal{L}_0\}$;
$\mathcal{L}$ is the line going through both $\mathcal{F}_{R_1}$ and $\mathcal{F}_{R_2}$, two distinct points;
$\mathcal{L}_0 \| \mathcal{L}$ and $s_0 \in \mathcal{L}_0$;
$\mathcal{L}_{p_1,p_2}$ is a line formed by a pair of antipodal poles p_1 and p_2;
$t_{\mathcal{F}}$ is a translation such that $t_{\mathcal{F}}(s_0) \in \mathcal{F}$.

Proof:

1. Assume $\mathcal{F} \neq \emptyset$.

For all nontrivial rotations $r \in R$, if there exists $x \in \mathcal{F}$ such that $r(x) \neq x$ then $r \notin R_1 \cap R_2 = R$, because x belongs to both $\mathcal{F}_{R_1}$ and $\mathcal{F}_{R_2}$. Thus $\mathcal{F} \subseteq \mathcal{F}_R$. Now let us choose $s'_0 \in \mathcal{F}, s'_0 = t_{\mathcal{F}}(s_0)$, then we have $R = t_{\mathcal{F}} R_c t_{\mathcal{F}}^{-1}, R_1 = t_{\mathcal{F}} R_{c_1} t_{\mathcal{F}}^{-1}, R_2 = t_{\mathcal{F}} R_{c_2} t_{\mathcal{F}}^{-1}$ (Proposition 4.50). This means that it is equivalent to discuss the intersection of pole-sets on S_0 or, alternatively, on S'_0 which is the translated unit sphere S_0 re-centered at s'_0 (Figure 4.7).

1.a $\mathcal{P}_R$

If $(p,n) \in \mathcal{P}_R$ then $|R^p_c| = n > 1$. From Proposition 4.10 and Proposition 4.11, $|R^p_c| = |R^{t_{\mathcal{F}}(p)}| = |R_1^{t_{\mathcal{F}}(p)} \cap R_2^{t_{\mathcal{F}}(p)}| = \gcd^*(|R_1^{t_{\mathcal{F}}(p)}|, |R_2^{t_{\mathcal{F}}(p)}|) = n > 1$ (Proposition 4.42). Using Proposition 4.10 once more, $\gcd^*(|R^p_{c_1}|, |R^p_{c_2}|) = n > 1$. Then $(p,n) \in \mathcal{P}_{R_1} \cap^* \mathcal{P}_{R_2}$ (Definition of $\mathcal{P}_{R_1} \cap^* \mathcal{P}_{R_2}$). Therefore $\mathcal{P}_R \subseteq \mathcal{P}_{R_1} \cap^* \mathcal{P}_{R_2}$.

If $(p,n) \in \mathcal{P}_{R_1} \cap^* \mathcal{P}_{R_2}$, then simply following the above arguments in reverse order one reaches the conclusion $(p,n) \in \mathcal{P}_R$. Thus $\mathcal{P}_{R_1} \cap^* \mathcal{P}_{R_2} \subseteq \mathcal{P}_R$.

This completes the proof that $\mathcal{P}_R = \mathcal{P}_{R_1} \cap^* \mathcal{P}_{R_2}$.

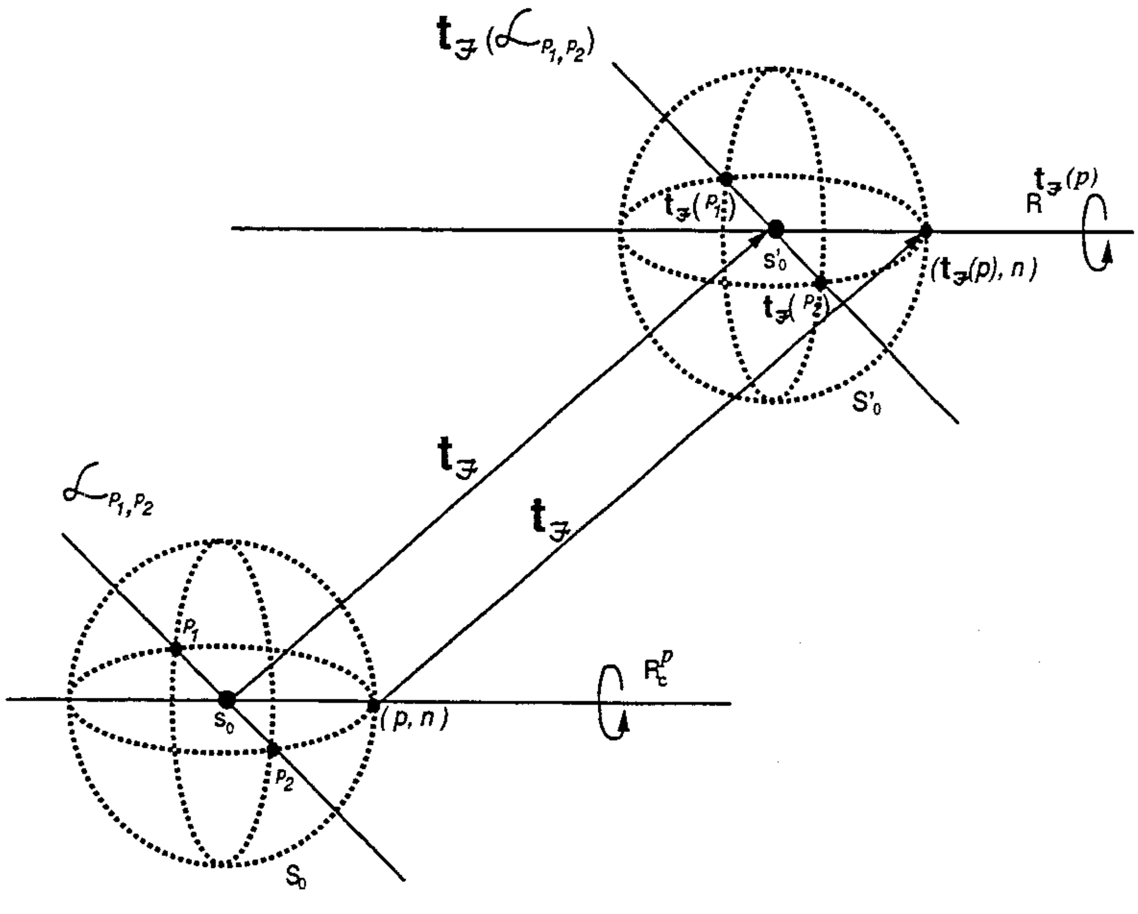

Figure 4.7: When $\mathcal{F} \neq \emptyset$, $R = t_{\mathcal{F}} R_c t_{\mathcal{F}}^{-1}$, $R_1 = t_{\mathcal{F}} R_{c_1} t_{\mathcal{F}}^{-1}$ and $R_2 = t_{\mathcal{F}} R_{c_2} t_{\mathcal{F}}^{-1}$

1.b $\mathcal{F}_R$

If $|\mathcal{P}_R| = 0$ then R is the identity group (Proposition 4.41). By Definition 4.8, $\mathcal{F}_R = \Re^3$. This is true for both $\mathcal{F} = \emptyset$ and $\mathcal{F} \neq \emptyset$.

If $|\mathcal{P}_R| = 2$ then R is a line group (Proposition 4.41). Thus $\mathcal{P}_R = \{(p_1, n), (p_2, n)\}$. $\mathcal{F}_{R_c}$ is a line containing p_1, p_2, and s_0. That is $\mathcal{F}_{R_c} = \mathcal{L}_{p_1,p_2}$. By Proposition 4.9 $\mathcal{F} = t_{\mathcal{F}}(\mathcal{F}_{R_c}) = t_{\mathcal{F}}(\mathcal{L}_{p_1,p_2})$ (Figure 4.7).

If $|\mathcal{P}_R| > 2$ then R is a point group (Proposition 4.41). Since $\mathcal{F} \neq \emptyset$, there exists at least one $x \in \mathcal{F}$, for all $r \in R, r(x) = x \Rightarrow x \in \mathcal{F}_R$. So $\mathcal{F} \subseteq \mathcal{F}_R$. However since R is a point group, $\mathcal{F}_R$ is a single point set. Hence $\mathcal{F} = \mathcal{F}_R$.

2. Assume $\mathcal{F} = \emptyset$.

If R_1, R_2 are both line groups, or R_1 is a line group and R_2 is a point group, then for all nontrivial rotations $r_1 \in R_1$, and all points $x \in \mathcal{F}_{R_2}, r_1(x) \neq x$. This is because a rotation only leaves points on its axis of rotation pointwise fixed. By the definition of the fixed point set (Definition 4.8), no such r_1 is a member of R_2. Therefore $R_1 \cap R_2 = \{1\} \Rightarrow \mathcal{P}_{R_1 \cap R_2} = \emptyset$. (We just completed the *otherwise* part for $\mathcal{P}_R$).

If R_1, R_2 are both point groups then for all nontrivial rotations r in R, $r(\mathcal{F}_{R_1}) = \mathcal{F}_{R_1}$ and $r(\mathcal{F}_{R_2}) = \mathcal{F}_{R_2}$. Therefore $r(\mathcal{L}) = \mathcal{L}$ where $\mathcal{L}$ is the line determined by the two distinct points $\mathcal{F}_{R_1}$ and $\mathcal{F}_{R_2}$ (Figure 4.8). By Definition 4.8 $\mathcal{F}_R = \mathcal{L}$. Therefore R will either be the identity group $\{1\}$ or a line group.

Now let $R = tR_ct^{-1}, R_1 = t_1R_{c_1}t_1^{-1}, R_2 = t_2R_{c_2}t_2^{-1}$. If R is a line group then $\mathcal{P}_R = \mathcal{P}_{R_c} = \{(p_1, n_1), (p_2, n_2)\}$ and $\mathcal{F}_{R_c}$ is a line determined by p_1, p_2 as shown in Figure 4.8.

For all $(p, n) \in \mathcal{P}_{R_c}$, $|R_c^p| = n > 1$. Then $|R_c^p| = |R^{t(p)}| = \gcd^*(|R_1^{t(p)}|, |R_2^{t(p)}|) = n > 1$. $R_1^{t(p)}$ fixes both $\mathcal{F}_{R_1}$ and $t(p)$, i.e. $\mathcal{L}$. Thus $R_1^{t(p)} = R_1^{tt'(p)} = R_1^{t_1(p)}$ and $R_2^{t(p)} = R_2^{tt''(p)} = R_2^{t_2(p)}$ (Figure 4.8), and so $\gcd^*(|R_1^{t_1(p)}|, |R_2^{t_2(p)}|) = \gcd^*(|R_{c_1}^p|, |R_{c_2}^p|) = n > 1$.

Since $s_0 \in \mathcal{F}_{R_c} = t^{-1}(\mathcal{F}_R) = t^{-1}(\mathcal{L}) = \mathcal{L}_0$ and $p \in \mathcal{F}_{R_c} = \mathcal{L}_0$, $p \in \mathcal{L}_0$. Thus $(p, n) \in \mathcal{P}_{R_1} \cap^*_{\mathcal{L}_0} \mathcal{P}_{R_2}$. Therefore $\mathcal{P}_R \subseteq (\mathcal{P}_{R_1} \cap^*_{\mathcal{L}_0} \mathcal{P}_{R_2})$.

For all $(p, n) \in \mathcal{P}_{R_1} \cap^*_{\mathcal{L}_0} \mathcal{P}_{R_2}$, following the above argument in reverse order one finds $(p, n) \in \mathcal{P}_R$. Thus $(\mathcal{P}_{R_1} \cap^*_{\mathcal{L}_0} \mathcal{P}_{R_2}) \subseteq \mathcal{P}_R$.

Therefore $\mathcal{P}_R = \mathcal{P}_{R_1} \cap^*_{\mathcal{L}_0} \mathcal{P}_{R_2}$ as claimed. □

This is a very important proposition which demonstrates the interplay between algebra and geometry. Compared to translations, rotations vary with respect to position in Euclidean space. This proposition shows that the varieties of rotations can be captured by simple and static elements, $\mathcal{P}$ and $\mathcal{F}$, such that the intersection

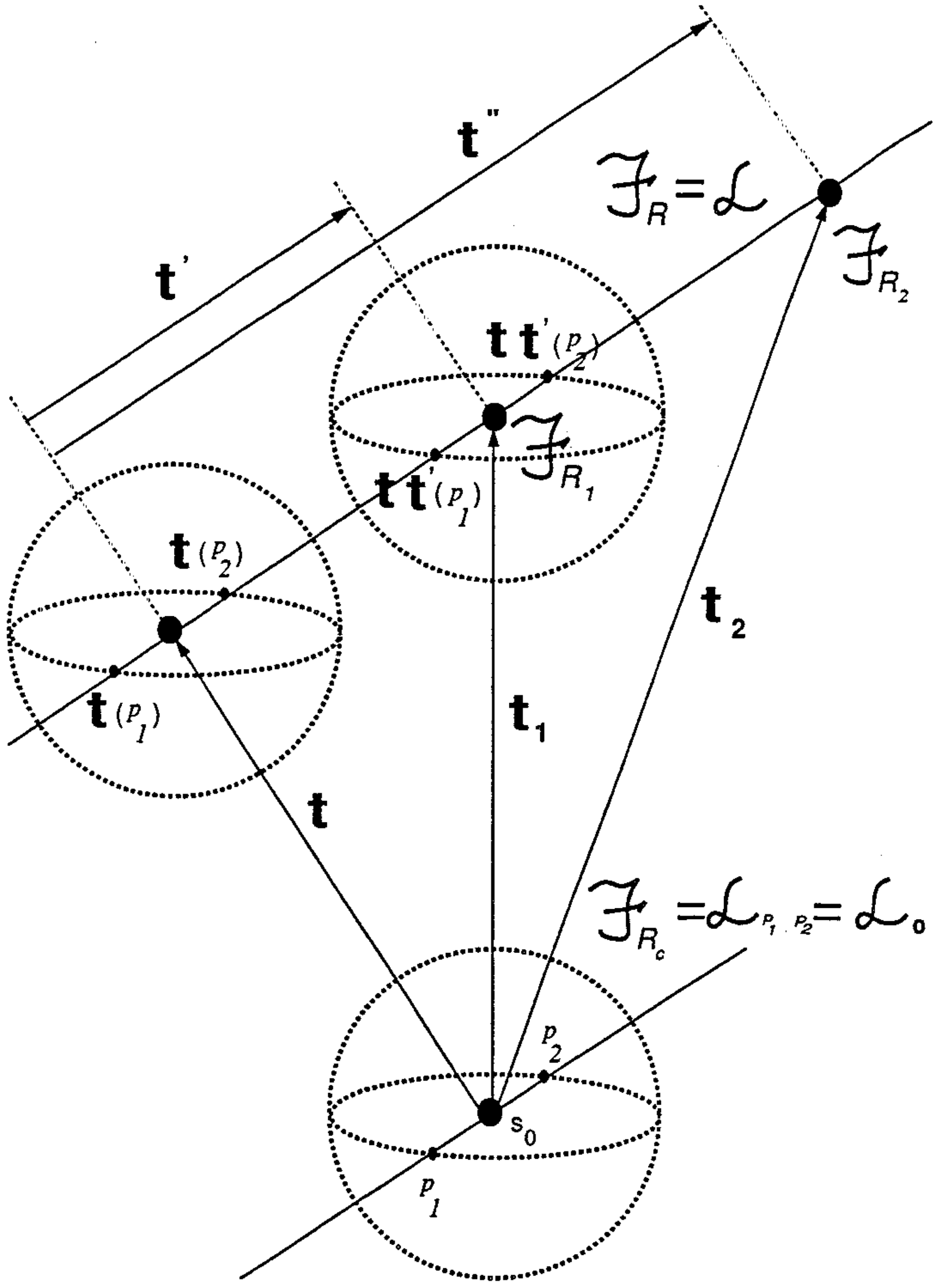

Figure 4.8: $R = R_1 \cap R_2$ is a line group. All the rotations in R fix both points $\mathcal{F}_{R_1}$ and $\mathcal{F}_{R_2}$. Thus they fix the line $\mathcal{L}$. So $\mathcal{F}_R = \mathcal{L}$.

computation of any rotation groups in $\Re^3$ can be performed at the origin s_0.

4.4 TR Subgroups of $\mathcal{E}^+$

Now comes to the point that translational invariants and rotational invariants are combined to provide the geometric characterization for the **TR** subgroups of $\mathcal{E}^+$.

Definition 4.54 TR subgroups of $\mathcal{E}^+$

A subgroup G of $\mathcal{E}^+$ is a **TR** *group iff $G = TR = \{tr | t \in T, r \in R\}$, where T is a translation subgroup and R is a rotation subgroup.*

In the following, let $G = TR$ be a **TR** group.

Lemma 4.55 $G \cap \mathbf{T}^3 = T$.

Proof:

If $g \in G \cap T^3$ then $g = tr$ must be a translation, with $t \in T, r \in R$. Then $t^{-1}g = r$ should be a translation also. However, for $r \in R$ to be both a rotation and a translation, there is only one possibility i.e. $r = 1$. Therefore $g = t \in T$. So $G \cap \mathbf{T}^3 \subseteq T$.

If $g \in T$ then g is a translation and $g \in G$, i.e. $g \in G \cap \mathbf{T}^3$. So $T \subseteq G \cap \mathbf{T}^3$. Therefore $G \cap \mathbf{T}^3 = T$. □

Lemma 4.56 *T is a normal subgroup of G, i.e. $T \lhd G$.*

Proof:

For all $r \in R$ and all $t \in T, rtr^{-1} \in T$, because rtr^{-1} is a translation (Proposition 4.25) and it is in G (Lemma 4.55). Then $rTr^{-1} \subseteq T$. For all $g = tr \in G$, since T^3 is an abelian group, $gTg^{-1} = trTr^{-1}t^{-1} = rTr^{-1} \subseteq T$. By the Definition of normal subgroup of a group (Definition 4.2), T is a normal subgroup of G. □

Proposition 4.57 *G is the semidirect product $T \times_\theta R$ where $\theta(r) : t \mapsto rtr^{-1}$.*

Proof:

Since T is a normal subgroup of G, $T \cap R = \{1\}$ and $G = TR$, G is a semidirect product of T and R (Proposition 4.3). □

Corollary 4.58 *For all $t \in T, TR = TtRt^{-1}$.*

Proof:

This proof uses the previous results namely $T \lhd G$ and $G = T \times_\theta R$.

For all $r \in R$ and for all $t \in T$ there exists $t' \in T$, such that $trt^{-1} = tt'r \in TR$, i.e. $tr = t'trt^{-1} \in TtRt^{-1}$. Therefore $\forall t \in T, TR \subseteq TtRt^{-1}$. Now for all $t', t \in T$ there exists $t'' \in T$ such that $t'trt^{-1} = t'tt''r \in TR$. Therefore $TtRt^{-1} \subseteq TR$. So we have $TtRt^{-1} = TR$. □

Proposition 4.59 *If $G = T_1R_1 = T_2R_2$ then $T_1 = T_2 = T$ and $R_1 = tR_2t^{-1}$ for some $t \in \mathbf{T}^3$.*

Proof:

By Lemma 4.55, $G \cap \mathbf{T}^3 = T_1 = T_2 = T$.

From Proposition 4.35 we have $R_1 = t_1R_{c_1}t_1^{-1}, R_2 = t_2R_{c_2}t_2{}^{-1}$ where R_{c_1}, R_{c_2} are subgroups of $SO(3)$, $t_1, t_2 \in \mathbf{T}^3$. Since $TR_1 = TR_2 \Rightarrow Tt_1R_{c_1}t_1{}^{-1} = Tt_2R_{c_2}t_2{}^{-1}$. For all $r_{c_1} \in R_{c_1}$ there exist $r_{c_2} \in R_{c_2}, t, t' \in T$ such that $tt_1r_{c_1}t_1^{-1} = t't_2r_{c_2}t_2^{-1} \Rightarrow tt_1r_{c_1}t_1^{-1}r_{c_1}^{-1}r_{c_1} = t't_2r_{c_2}t_2^{-1}r_{c_2}^{-1}r_{c_2}$. Since both $r_{c_1}t_1r_{c_1}^{-1}, r_{c_2}t_2r_{c_2}^{-1} \in \mathbf{T}^3$ (Proposition 4.25), by Proposition 4.24 we can equate the rotational elements of both sides $r_{c_1} = r_{c_2} \in R_{c_2}$. Thus $R_{c_1} \subseteq R_{c_2}$. Similarly, $R_{c_2} \subseteq R_{c_1}$. Therefore $R_{c_1} = R_{c_2} \Rightarrow \mathcal{P}_{R_{c_1}} = \mathcal{P}_{R_{c_2}}$ (Proposition 4.50). By Proposition 4.49 we have $\mathcal{P}_{R_1} = \mathcal{P}_{R_2}$ and $\exists t \in \mathbf{T}^3$ such that $R_1 = tR_2t^{-1}$. □

This proposition states that the representation of a **TR** group in terms of its maximal translation and rotation subgroups is unique up to the conjugation of its rotation subgroups by a translation t in $\mathbf{T}^3$. Since Corollary 4.58 shows that in this representation R can be conjugated via all the $t \in T$, the question to be asked now is whether t can be outside of T. In the next section we shall see that if T is a continuous translation subgroup then $t \in T$, otherwise t is not necessarily in T but in a form derived from the generator(s) of T.

4.4.1 Translational Characteristic Invariant for TR subgroups

From the previous proposition we can conclude that since T is unique in G, we can describe all the translations in G by characterizing T only. Therefore we have the following definition:

Definition 4.60 *If $G = TR$ is a* **TR** *group then the* **translational invariant** $\mathcal{T}_G$ *of G is defined to be the T-orbit of the origin s_0.*

$$\mathcal{T}_G = T(s_0) = \{t(s_0) | t \in T\}$$

Let $G = TR$ and $R = tR_ct^{-1}$ where $t \in \mathrm{T}^3, R_c \subseteq SO(3)$.

Proposition 4.61 *Let* $R = tR_ct^{-1}, t \in \mathrm{T}^3$. $G = TR$ *is a group* $\Leftrightarrow^{(1)} \forall r \in R, r(t(\mathcal{T}_G)) = t(\mathcal{T}_G) \Leftrightarrow^{(2)} \forall r_c \in R_c, r_c(\mathcal{T}_G) = \mathcal{T}_G$. *This property is called the* **TR-restriction**.

Proof :

For each $r \in R$ there exists $r_c \in R_c$ such that $r = tr_ct^{-1}$. Then $r(t(s_0)) = tr_ct^{-1}(t(s_0)) = tr_c(s_0) = t(s_0)$.

($\Rightarrow^{(1)}$) If G is a **TR** group then for all $r \in R$,

$$
\begin{array}{rll}
rt(\mathcal{T}_G) & = r(t(T(s_0)) & \textit{Definition } 4.60 \\
& = rT(t(s_0)) & \textit{commutativity of translations} \\
& = rTr^{-1}(t(s_0)) & \textit{since } t(s_0) = r(t(s_0)) \\
& = Tt(s_0) & \textit{Since } T \lhd G \\
& = t(T(s_0)) & \textit{commutativity of translations} \\
& = t(\mathcal{T}_G) & \textit{Definition } 4.60
\end{array}
$$

($\Rightarrow^{(2)}$) If $\forall r \in R, r(t(\mathcal{T}_G)) = t(\mathcal{T}_G)$ then $\exists r_c \in R_c$ such that $r(t\mathcal{T}_G) = tr_ct^{-1}(t\mathcal{T}_G) = tr_c(\mathcal{T}_G) = t(\mathcal{T}_G) \Rightarrow r_c(\mathcal{T}_G) = \mathcal{T}_G$.

($\Leftarrow^{(1)}$) Since for all $r \in R, r(t(\mathcal{T}_G)) = t(\mathcal{T}_G)$ then $rt(T(s_0)) = tT(s_0) \Rightarrow T(t(s_0)) = rT(t(s_0)) = rTr^{-1}(t(s_0))$. For all $t_r \in rTr^{-1}, \exists t' \in T$ such that $t_r(t(s_0)) = t'(t(s_0))$. By Proposition 4.29 $t_r = t' \Rightarrow rTr^{-1} \subseteq T$. Similarly, we can prove $T \subseteq rTr^{-1}$. Therefore $rTr^{-1} = T$. It can be easily verified that TR forms a group with T as its normal subgroup. By Proposition 4.57 $G = TR$ is a **TR** group.

($\Leftarrow^{(2)}$) If $\forall r_c \in R_c, r_c(\mathcal{T}_G) = \mathcal{T}_G$ then $r_c(\mathcal{T}_G) = t^{-1}rt(\mathcal{T}_G) = \mathcal{T}_G \Rightarrow r(\mathcal{T})) = t(\mathcal{T}_G)$.
□

Since in a **TR** group $G = TR$, T can be discrete as well as continuous, the choices for $\mathcal{T}_G$ include planes, lines, parallel lines, parallel planes, one dimension lattices and two dimension lattices[4]. When $\mathcal{T}_G$ is a one or two dimensional discrete grid (lattice), saying a rotation r is parallel or perpendicular to such a $\mathcal{T}_G$ means that $\bar{r}$ is parallel or perpendicular to the minimal manifold containing $\mathcal{T}_G$.

Corollary 4.62 *If* $\Re^3 \neq \mathcal{T}_G \neq \emptyset$ *and* $|\mathcal{P}_R| > 0$ *then for all* $r \in R$, *either* $\bar{r} \| t(\mathcal{T}_G)$ *or* $\bar{r} \perp t(\mathcal{T}_G)$

[4]Although the three dimensional discrete translation group satisfies all the previous definitions and properties for **TR** groups, from now on we exclude it in our proofs since it is a special case for our purposes.

Proof:

It is clear that for the finite choices of $\mathcal{T}_G$ listed above the only nontrivial rotations $r \in R$ such that $r(t(\mathcal{T}_G)) = t(\mathcal{T}_G)$ have to be $\bar{r}\|t(\mathcal{T}_G)$ or $\bar{r}\perp t(\mathcal{T}_G)$ (Proposition 4.61). Similarly, we have $\forall r_c \in R_c, \bar{r}_c\|\mathcal{T}_G$ or $\bar{r}_c\perp\mathcal{T}_G$. □

Lemma 4.63 *If* $G = TR$ *is a* **TR** *group and* $R = tR_ct^{-1}$, $R_c \subseteq SO(3), t \in \mathbf{T}^3$ *then* $\mathcal{F}_R \cap t(\mathcal{T}_G) \neq \emptyset$.

Proof:

From the proof of Lemma 4.61 we have $t(s_0) \in \mathcal{F}_R$, since $s_0 \in \mathcal{T}_G \Rightarrow t(s_0) \in t(\mathcal{T}_G) \Rightarrow \mathcal{F}_R \cap t(\mathcal{T}_G) \neq \emptyset$. □

Proposition 4.64 *If* $G = TR$ *is a* **TR** *group,* T *is continuous, and* $g \in G$ *is a nontrivial rotation, then* $g \in tRt^{-1}$, *for some* $t \in T$.

Proof:

Since $g = t_g r$ for some $t_g \in T, r \in R$ and g is a rotation in G, then

1) $\bar{g}\|\bar{r}$ and $\theta_g = \theta_r$ (Corollary 4.32).

2) Let $R = t_cR_ct_c^{-1}$. By the **TR**-restriction $r(t_c(\mathcal{T}_G)) = t_c(\mathcal{T}_G)$. Since $t_g \in T$, $g(t_c(\mathcal{T}_G)) = t_g r(t_c(\mathcal{T}_G)) = t_g t_c(\mathcal{T}_G) = t_c t_g(\mathcal{T}_G) = t_c(\mathcal{T}_G)$. Since T is continuous, $\mathcal{T}_G$ is a line or a plane. It must be true that $\bar{r} \cap t_c(\mathcal{T}_G) \neq \emptyset$ and $\bar{g} \cap t_c(\mathcal{T}_G) \neq \emptyset$.

Given the above constraints on g and r, let us look into each case of $\mathcal{T}_G$ and r:

- $\mathcal{T}_G$ is a line and $\bar{r}\|\mathcal{T}_G$. Since $\bar{g}\|\bar{r}, \bar{g}\|\mathcal{T}_G$. Since $g(t_c\mathcal{T}_G) = t_c(\mathcal{T}_G), \bar{g} = t_c(\mathcal{T}_G)$. Therefore $\bar{r} = \bar{g} \Rightarrow r = g \in R$.

- $\mathcal{T}_G$ is a line or a plane and $\bar{r}\perp\mathcal{T}_G$. Then $\bar{g}\perp\mathcal{T}_G$. Since $\bar{g} \cap t_c(\mathcal{T}_G) \neq \emptyset$ and T acts on $t_c(\mathcal{T}_G)$ transitively, there always exists $t \in T$ such that $\bar{g} = t(\bar{r})$. By Corollary 4.32 $g = trt^{-1} \in tRt^{-1}$.

- $\mathcal{T}_G$ is a plane and $\bar{r}\|\mathcal{T}_G$. So $\bar{g}\|\mathcal{T}_G$. Since $g(t_c\mathcal{T}_G) = t_c(\mathcal{T}_G)$ and $\mathcal{T}_G$ is continuous, it must be true that $\bar{g} \subset t_c(\mathcal{T}_G)$, and also $\bar{r} \subset t_c(\mathcal{T}_G)$. Since T is continuous and T acts on $t_c(\mathcal{T}_G)$ transitively, there exists $t \in T$ such that $\bar{g} = t(\bar{r})$. By Corollary 4.32 $g = trt^{-1} \in tRt^{-1}$. □

This proposition proves that when T is continuous any rotation in a **TR** group G must belong to some rotation group R' where $G = TR'$. In another words, the set $\{trt^{-1} | t \in T, r \in R\}$ contains *all* the possible rotations in G.

Proposition 4.65 *Given $G = TR_1 = TR_2$, if T is a continuous translation group then there exists $t \in T$ such that $R_1 = tR_2t^{-1}$.*

Proof:

If $T \cong \mathbf{T}^3$ then any translation t such that $R_1 = tR_2t^{-1}$, $t \in T$. If $T = \{1\}$ then $R_1 = R_2$.

Now we consider $T \cong \mathbf{T}^1$ and $T \cong \mathbf{T}^2$ only.

From Proposition 4.64, we know that if $g \in G$ is a nontrivial rotation then $g \in t_1R_1t_1^{-1}$ and $g \in t_2R_2t_2^{-1}$ where $t_1, t_2 \in T$. Since by Proposition 4.59 there exists $t \in \mathbf{T}^3$ such that $R_1 = tR_2t^{-1}$, $\mathcal{F}_{R_1}, \mathcal{F}_{R_2}$ are either both lines or both points of $\Re^3$.

If R_1, R_2 are both line groups then $\bar{g} = \mathcal{F}_{t_1R_1t_1^{-1}} = \mathcal{F}_{t_2R_2t_2^{-1}} = t_1(\mathcal{F}_{R_1}) = t_2(\mathcal{F}_{R_2})$. Thus $\mathcal{F}_{R_1} = t_1^{-1}t_2(\mathcal{F}_{R_2})$ and so $R_1 = t_1^{-1}t_2R_2t_2^{-1}t_1 = tR_2t^{-1}$ (Proposition 4.52).

If R_1, R_2 are point groups then $t_1(\mathcal{F}_{R_1}) \in \bar{g}$ and $t_2(\mathcal{F}_{R_2}) \in \bar{g}$. By the **TR**-restriction there are only two choices for $\bar{g}$, i.e. $\bar{g}\|\mathcal{T}_G$ or $\bar{g}\perp\mathcal{T}_G$:

When $\bar{g}\|\mathcal{T}_G$, since T is continuous there exists $t' \in T$ such that $\mathcal{F}_{t_1R_1t_1^{-1}} = t'(\mathcal{F}_{t_2R_2t_2^{-1}})$. Therefore $t_1(\mathcal{F}_{R_1}) = t't_2(\mathcal{F}_{R_2}) \Rightarrow R_1 = tR_2t^{-1}, t = t_1^{-1}t't_2 \in T$ (Proposition 4.52).

When $\bar{g}\perp\mathcal{T}_G$, assuming there exists $t_\perp \perp \mathcal{T}_G$ and $t_\perp \neq 1$ such that $t_1(\mathcal{F}_{R_1}) = t_\perp t_2(\mathcal{F}_{R_2})$ (Figure 4.9). Obviously $t_\perp \notin T$. Let $R_1' = t_1R_1t_1^{-1}, R_2' = t_2R_2t_2^{-1}$. For each nontrivial rotation $r_1 \in R_1'$ there exists $r_2 \in R_2'$ such that $r_1 = t_\perp r_2 t_\perp^{-1} \Rightarrow r_1r_2^{-1} = t_\perp(r_2t_\perp^{-1}r_2^{-1})$. Thus both sides are tranlations in T (Lemma 4.55). If $\mathcal{T}_G$ is a line and $r_1\perp\mathcal{T}_G$ (since R_1, R_2 are point groups), or $\mathcal{T}_G$ is a plane where r_1 has to be $r_1\|\mathcal{T}_G$, then $\theta_{r_1} = \theta_{R_2}$ has to be π (TR-restriction). Then $t_\perp(r_2t_\perp^{-1}r_2^{-1}) = 2t_\perp \notin T$, a contradiction. If $\mathcal{T}_G$ is a line and $r_1\|\mathcal{T}_G$ then $t_\perp(r_2t_\perp^{-1}r_2^{-1})\perp T$ and thus $\notin T$ unless $\theta_{r_1} = \theta_{r_2} = 0$, a contradiction. Therefore, $r_\perp = 1$ and so $t_1(\mathcal{F}_{R_1}) = t_2(\mathcal{F}_{R_2}) \Rightarrow \mathcal{F}_{R_1} = t_1^{-1}t_2(\mathcal{F}_{R_2})$. By Proposition 4.52 $R_1 = t_1^{-1}t_2R_2t_2^{-1}t_1 = tR_2t^{-1}$ where $t = t_1^{-1}t_2 \in T$.
□

When T is a discrete translation group, there are cases where g is a rotation in G, but $g \notin R_i$ for any rotation subgroup R_i such that $G = TR_i$. Therefore it is not sufficient to characterize R_i only. Figure 4.10 taken from [75] illustrates the unit lattices of seventeen two dimensional Crystallographic groups. A diamond shaped symbol denotes a 2-fold rotation symmetry, a square denotes a 4-fold rotation etc. The Crystallographic group p_4, the symmetry group of a two-dimensional lattice generated by a square lattice unit, is an example in which the two-fold rotations about the mid-point of each edge of a unit lattice are in none of the rotation subgroups R_i of $G = TR_i$, rather, they are products from distinct R_is.

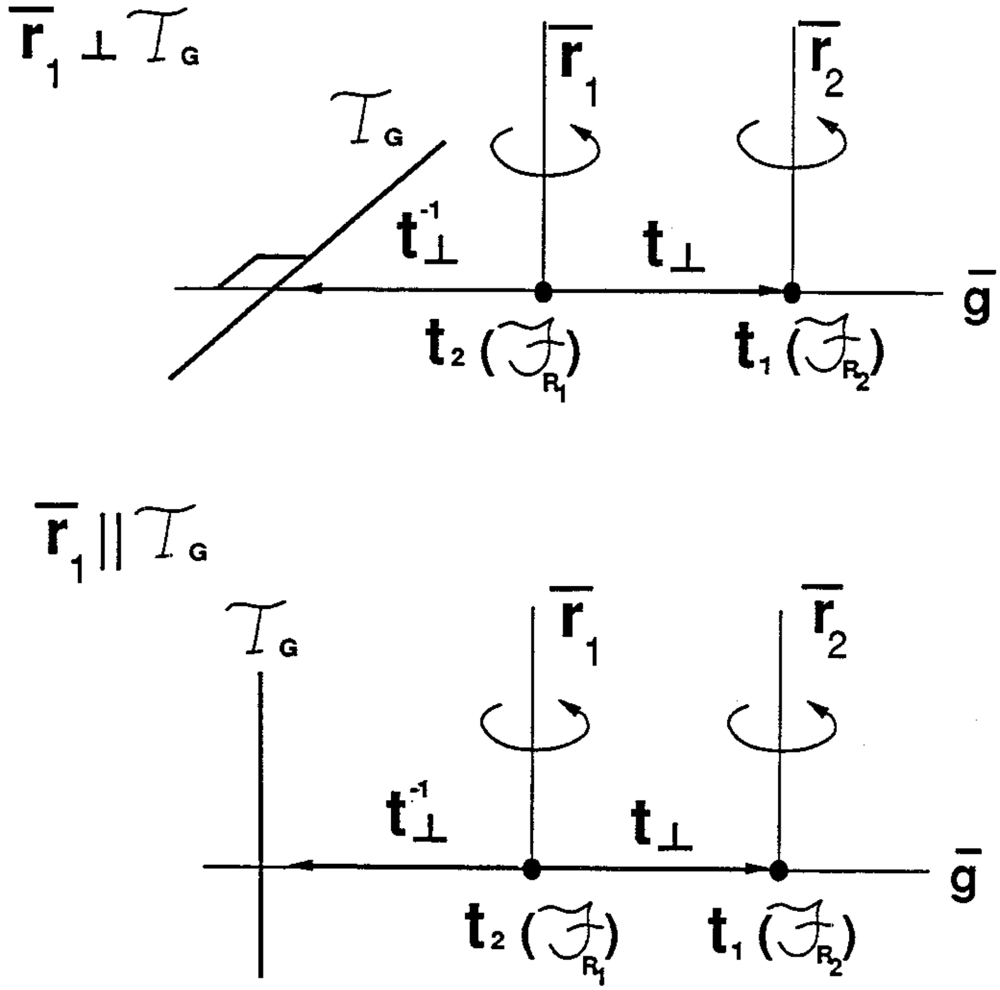

Figure 4.9: When R_1, R_2 are both point groups and $\bar{g} \perp \mathcal{T}_G$

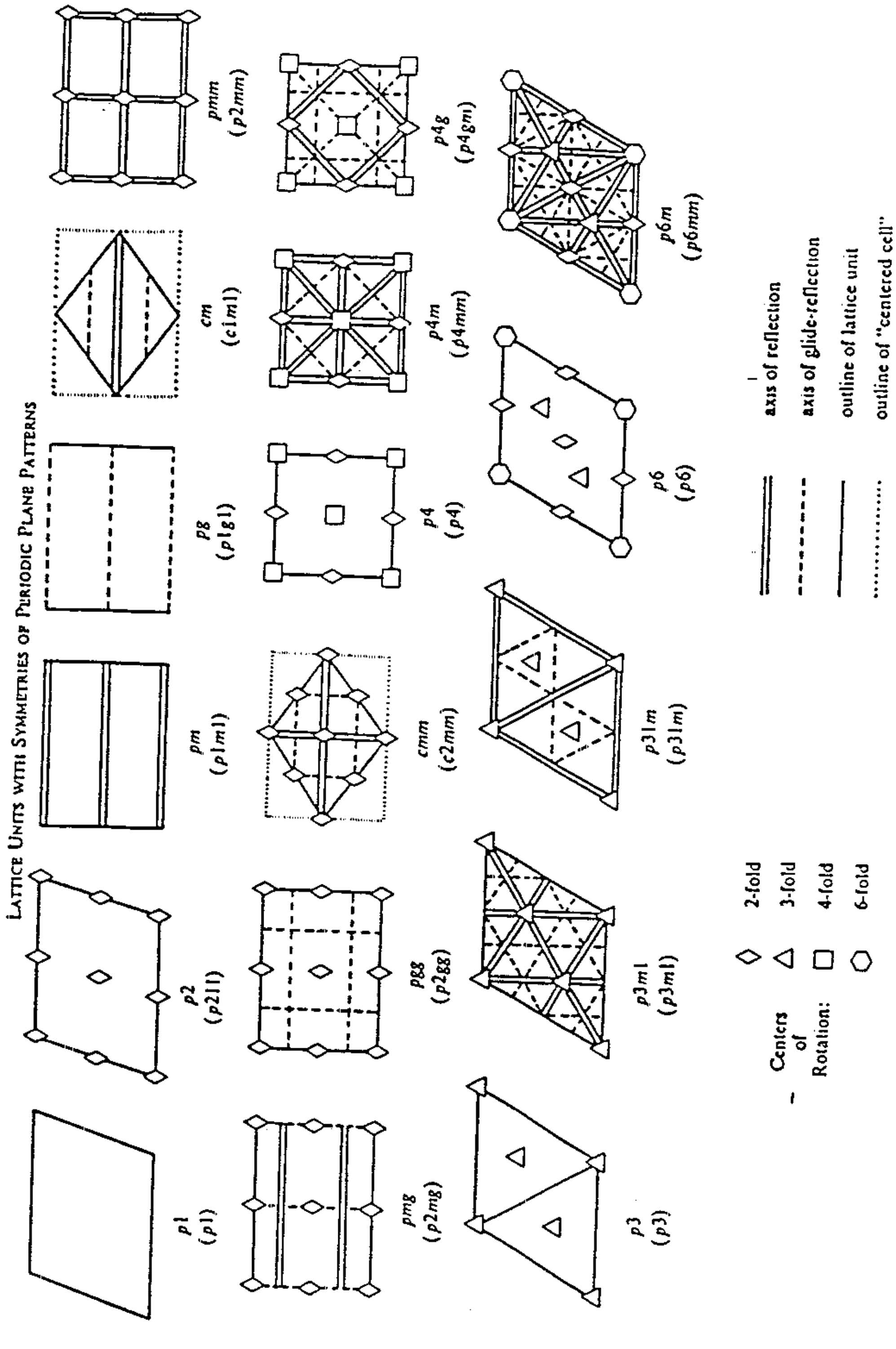

Figure 4.10: The seventeen two-dimensional Crystallographic groups

4.4.2 Crystallographic Groups

The observation in the previous section on discrete **TR** groups calls for a more elaborate characterisation for the rotational symmetry of Crystallographic groups. Fortunately, such groups have been carefully studied (see [40, 75, 76]). A *Frieze* is a pattern which is invariant under all multiples of just one translation. The associated symmetry groups (the set of all isometries which map the pattern onto itself [75]) are commonly called **frieze groups**. Patterns that are invariant under linear combinations of two linearly independent translations repeat at regular intervals in two directions and their symmetry groups are often termed **wallpaper groups** or plane groups. Similarly, if there are three independent translations, the associated symmetry groups are called **space groups.** These three types of groups form the basic set of Crystallographic groups. Due to the famous *Crystallographic restriction*, the only possible n-fold rotational symmetries in Crystallographic groups are when $n = 1, 2, 3, 4, 6$. Only a *finite* number of essentially distinct Crystallographic groups exist. There are *seven* frieze groups in a plane, *seventeen* wallpaper groups, and 230 space groups [27, 32].

Since we are interested in those symmetry groups that act on a pattern in the three dimension Euclidean space, the behavior of the Frieze groups and the plane groups should be re-examined in 3-space. These groups have been traditionally studied on a static plane as planar mixed polarity groups or black-white groups [76]. **TR** groups only include those Crystallographic groups which can be represented as a semidirect product of a translation subgroup T and a rotation subgroup R. Those Crystallographic groups with reflections, glide-reflections and glide-rotations as generators are excluded. Table 4.1 gives all the **TR** Crystallographic groups from an exhaustive counting of the $7 + 17 = 24$ one-dimensional two-sided infinite Crystallographic groups and $17 + 46 = 63$ two-dimensional two-sided infinite Crystallographic groups[5]. We have omitted those black-white groups which have the same color on both sides because they have the principal-plane reflection symmetries as their generators. Therefore the total number of one and two dimensional **TR** Crystallographic groups is $4 + 12 = 16$. Space Crystallographic groups contribute to the **TR** groups 12 more besides the 12 plane ones, giving a total of 24 (Table 4.2).

In [75] seventeen plane Crystallographic groups are discussed. Figure 4.10 from

[5]Note, three out of the four one-dimensional **TR** Crystallographic groups in Table 4.1 have the same labelings as some members of the two-dimensional **TR** Crystallographic groups, however they are different groups.

Table 4.1: The **TR** subgroups in one and two dimensional Crystallographic Groups

Crystallographic Groups	Freize Group ($1D$)	Wallpaper Group ($2D$)
Number of Polar ones (on a fixed plane)	7	17
The **TR** ones	$p1, p112$	$p1, p2, p3, p4, p6$
Mixed polar ones (on a flipping plane)	17	46
The **TR** ones	$p121, p211, p222$	$p121, c121, p222, c222$ $p422, p312, p321, p622$
Notes	$p121 = p112$	$p2$ has two forms $p112 = p121$
Total in **TR**	4	$5 + 7 = 12$

Table 4.2: The **TR** subgroups in three dimensional Crystallographic Groups

Crystallographic Group	Space groups ($3D$)
Total Number	230
The **TR** ones	$P432, P23, B112, F23, F222$ $I222, I23, I4, I422, R3, F432, I432$
Notes	plus all the 2D ones by adding one translation $\perp$ to the plane
Total in **TR**	24

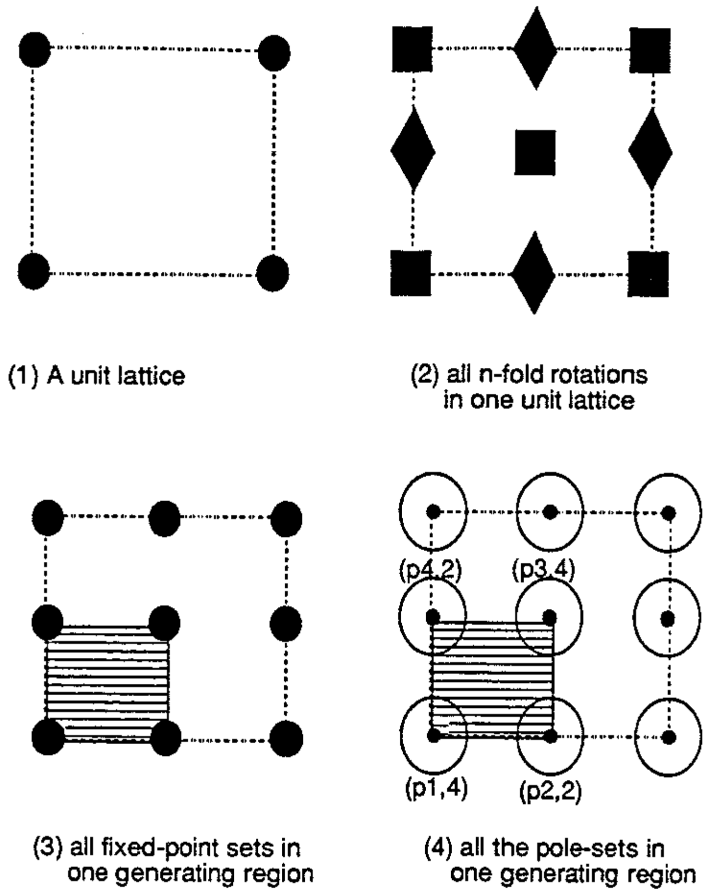

Figure 4.11: The Crystallographic group $p4$ characterised by $\mathcal{F}_{\hat{R}}$ and $\mathcal{P}_{\hat{R}}$.

[75] shows the complete symmetries for each *primitive cell* (lattice unit or unit cell) of the seventeen possible one-sided plane patterns. Usually a primitive cell is chosen with centers of highest order of rotation at the vertices. A set of figures similar to Figure 4.10 can be given for the Freize groups as well [77]. All the possible rotations can be represented explicitly in a primitive cell and then translated all over the space/plane/line. Furthermore, rotations in each primitive cell can be generated by a *generating region* (also called asymmetric unit, fundamental region or fundamental domain) which can be even smaller than the primitive cell itself. A generating region is the smallest region of the plane (line) whose images under the full symmetry group of the lattice cover the whole plane (line). For example, Figure 4.11 shows that only a quarter of the primitive cell is chosen as the generating region of $p4$, one of the wallpaper groups. In order to capture all the rotations in a **TR** Crystallographic group $G = TR$ we need to make those rotations of G which neither belong to R nor

to any conjugate of R through $t \in T$ explicit. Since we use a fixed point set and a pole-set to represent a rotation group, one obvious way to achieve this is to extend the pole-set $\mathcal{P}_R$ and fixed point set $\mathcal{F}_R$ for a single rotation group R to those for a set of rotation subgroups within the generating region of, $\mathcal{F}_{all}$, the fixed point sets of all the rotation subgroups in TR. $\mathcal{F}_{all}$ is usually denser than the corresponding translational invariant.

According to the convention of primitive cell in a Crystallographic group $G = TR$, R always appears at a corner of a unit lattice. Let $\tilde{R}$ denote the set of rotation subgroups in the generating region of which R is located at the lower left corner:

$$\mathcal{F}_{\tilde{R}} = \{\mathcal{F}_{R_1}, \ldots, \mathcal{F}_{R_n}\}, \mathcal{P}_{\tilde{R}} = \{\mathcal{P}_{R_1}, \ldots, \mathcal{P}_{R_n}\}$$

The essence of number n here is that $\mathcal{F}_{\tilde{R}}$ contains those and only those points (lines) which are not in the same T orbit of $\mathcal{F}_{all} = T(\mathcal{F}_{\tilde{R}})$. In the one dimensional case there are only two distinct T orbits $n = 2$. There are four distinct T orbits in the generating region of $p4 = TR$ therefore $\mathcal{F}_{\tilde{R}}$ contains four fixed point sets and $\mathcal{P}_{\tilde{R}}$ contains four pole-sets correspondingly (Figure 4 11). Since the number of Crystallographic groups is finite and their $\mathcal{F}_{all}$ are known, $\mathcal{F}_{\tilde{R}}$ can be determined constructively.

Lemma 4.66 *If $g \in G = TR$ is a rotation and T is discrete then $g \in t\tilde{R}t^{-1}$ for some $t \in T$*

Proof:

If $g \in G = TR$ is a rotation then $\bar{g} \cap T(\mathcal{F}_{\tilde{R}}) \neq \emptyset$ (by the definition of $T(\mathcal{F}_{\tilde{R}})$). Then there exists at least one $t \in T$ such that $\bar{g} \cap t(\mathcal{F}_{\tilde{R}}) \neq \emptyset$. If $g \notin t\tilde{R}t^{-1}$ then $g \notin G$, a contradiction. Therefore $g \in t\tilde{R}t^{-1}$. □

Definition 4.67 *A group T* **acts on a discrete set of points D transitively in terms of** *$\mathcal{F}$ means that if for $S_1, S_2 \subset D$, there exist $t_1, t_2 \in \mathrm{T}^3$ such that $S_1 = t_1(\mathcal{F}), S_2 = t_2(\mathcal{F})$, then there exists $t \in T$ such that $S_1 = t(S_2)$.*

Let $t' = 1$ or

$$t' = \sum \frac{t_i}{n} \tag{4.2}$$

Here $\sum$ denotes the collective composition of translations, t_i represents the discrete generator(s) of T, and $n = 2$ or 3 depending on which of the 16 **TR** Crystallographic groups together with the parallel-line and parallel-plane groups is under consideration.

Fact: If $G = TR$, and T is discrete, then $\mathcal{F}_{\tilde{R}}$ and t' can be so chosen such that the group $< t_i, t' >$ acts on $\mathcal{F}_{all} = T(\mathcal{F}_{\tilde{R}})$ transitively in terms of $\mathcal{F}_{\tilde{R}}$.

In the next two lemmas and one proposition, t' represents the translation defined in equation 4.2.

Lemma 4.68 *If $G = T_1R_1 = T_2R_2$ and T is discrete then $R_1 = tt'R_2(tt')^{-1}$ where $t \in T$.*

Proof:

Using the above fact, there exist $t \in T$ and t' defined in equation 4.2 such that $\mathcal{F}_{\tilde{R}_1} = tt'(\mathcal{F}_{\tilde{R}_2}) \Rightarrow \mathcal{F}_{R_1} = \mathcal{F}_{R_2}$. Thus $R_1 = tt'R_2(tt')^{-1}$. □

Lemma 4.69 *If $G = TR$ and T is discrete then $T(\mathcal{F}_{\tilde{R}}) = T(t'(\mathcal{F}_{\tilde{R}}))$.*

Proof:

When $t' = 1$, $T(\mathcal{F}_{\tilde{R}}) = T(t'(\mathcal{F}_{\tilde{R}}))$.

Suppose T is a one dimensional discrete translation group with generator t_0. G has $\mathcal{F}_{\tilde{R}} = \{a, b\}$ for some $\{a, b\}$ for which $a = t'(b)$ where $t' = \frac{1}{2}t_0$. Hence $t'^2 \in T$. The T orbit of the fixed point set is $T(\mathcal{F}_{\tilde{R}}) = T(\{a, b\}) = T(\{a, t'(a)\})$. Here $\{T(a)\}$ and $\{T(b)\} = \{t'T(a)\}$ are two disjoint T orbits. Now, $T(t'(\mathcal{F}_{\tilde{R}}) = T(\{t'(a), t'(b)\}) = \{t'T(a)\} \cup \{Tt't'(a)\} = \{t'T(a)\} \cup \{T(a)\} = T(\{t'(a), a\}) = T(\{b, a\}) = T(\mathcal{F}_{\tilde{R}})$.

When T is a two dimensional discrete translation group the proof can be carried out in a similar manner. For some of these groups t' is always the identity. For example, $p6 = TR_1 = TR_2$ then $R_1 = tR_2t^{-1}, t \in T$. For others such as $p3$ and $p4$, there are cases $t' \neq 1$, and there can be more than one form ($p3$ has $n = 3$ and there are two different forms of t'). For each form of t', it can be shown that $T(\mathcal{F}_{\tilde{R}}) = T(t'(\mathcal{F}_{\tilde{R}}))$. □

Proposition 4.70 *If $G = TR_1 = TR_2$ then $T(\mathcal{F}_{R_1}) = T(\mathcal{F}_{R_2})$.*

Proof:

When T is continuous, there exists $t \in T$, such that $R_1 = tR_2t^{-1}$ (Proposition 4.65). By Proposition 4.9, $\mathcal{F}_{R_1} = t\mathcal{F}_{R_2} \Rightarrow T(\mathcal{F}_{R_1}) = T(t\mathcal{F}_{R_2}) = T(\mathcal{F}_{R_2})$.

When T is discrete, $\mathcal{F}_{R_1}$ represents the fixed point sets in the *generating region* denoted by R_1, as does $\mathcal{F}_{R_2}$. By Lemma 4.68 and Proposition 4.9, $\mathcal{F}_{\tilde{R}_1} = tt'(\mathcal{F}_{\tilde{R}_2})$ where $t \in T$. Then $T(tt'(\mathcal{F}_{\tilde{R}_2})) = T(t'(\mathcal{F}_{\tilde{R}_2})) = T(\mathcal{F}_{\tilde{R}_2})$ (Lemma 4.69). Therefore $T(\mathcal{F}_{\tilde{R}_1}) = T(\mathcal{F}_{\tilde{R}_2})$. □

This leads us to the following definition.

4.4.3 Rotational Characteristic Invariants for TR subgroups

Definition 4.71 *The* **rotational invariant of a TR group** $G = TR$ *is defined to be the pair*

$$(\mathcal{F}_G, \mathcal{P}_G) = (T(\mathcal{F}_R), \mathcal{P}_R)$$

When T is a discrete group, $\mathcal{F}_R$ and $\mathcal{P}_R$ should be understood as the **fixed point set** and **pole-sets** of the rotation subgroups in the generating region denoted by R. This interpretation is actually a more general concept since when T becomes continuous, i.e. the distance between the discrete lattice points becomes infinitely small, a pole structure, usually bearing several pole-pierced unit spheres, shrinks into one unit sphere at the origin.

This definition differs from that of rotational characteristic invariants for rotation groups in that the fixed-point set is moved around in $\Re^3$ by the translation subgroup of G. For example, the rotation symmetry group of an infinite cylinder $G_{cyl} = T_1 R_{cyl}$ has rotations about the central axis as well as 180° rotations about any axis perpendicular to and intersecting with the central axis. These rotations are not restricted to be around s_0 only but can 'take a ride' on the translations in G to anywhere in $T(\mathcal{F}_R)$, as shown in Figure 4.12 (compare with Figure 4.6).

Next we prove the one-to-one correspondence between the **TR** groups and their characteristic invariants just defined.

Theorem 4.1 $f : \{G = TR\} \rightarrow \{(\mathcal{T}_G, (\mathcal{F}_G, \mathcal{P}_G))\}$ *is an injection.*

Proof:

Let $G_1 = T_1 R_1, G_2 = T_2 R_2$. Given $(\mathcal{T}_{G_1}, (\mathcal{F}_{G_1}, \mathcal{P}_{G_1})) = (\mathcal{T}_{G_2}, (\mathcal{F}_{G_2}, \mathcal{P}_{G_2}))$; we are going to prove that $G_1 = G_2$.

Since $\mathcal{T}_{G_1} = \mathcal{T}_{G_2}$, for all $t_1 \in T_1, t_1(\mathcal{T}_{G_2}) = \mathcal{T}_{G_2}$. By Proposition 4.30 $t_1 \in T_2$. Thus $T_1 \subseteq T_2$. Similarly, $T_2 \subseteq T_1$. Therefore $T_1 = T_2 = T$.

Since $\mathcal{P}_{R_1} = \mathcal{P}_{R_2}$, by Proposition 4.49, there exists a translation t_0 such that $R_1 = t_0 R_2 t_0^{-1}$. So $TR_1 = Tt_0 R_2 t_0^{-1} = t_0(TR_2)t_0^{-1}$. We have shown that G_1 and G_2 are at least conjugate with each other.

When T is a continuous subgroup of $\mathbf{T}^3$, $\mathcal{F}_{R_1}, \mathcal{F}_{R_2}$ are either two points or two parallel lines in $\Re^3$. From $T(\mathcal{F}_{R_1}) = T(\mathcal{F}_{R_2})$, it follows that there exists $t \in T$ such that $\mathcal{F}_{R_1} = t(\mathcal{F}_{R_2})$. By Proposition 4.50 $R_1 = tR_2t^{-1}$. By Corollary 4.58 $G_1 = TR_1 = TtR_2t^{-1} = TR_2 = G_2$.

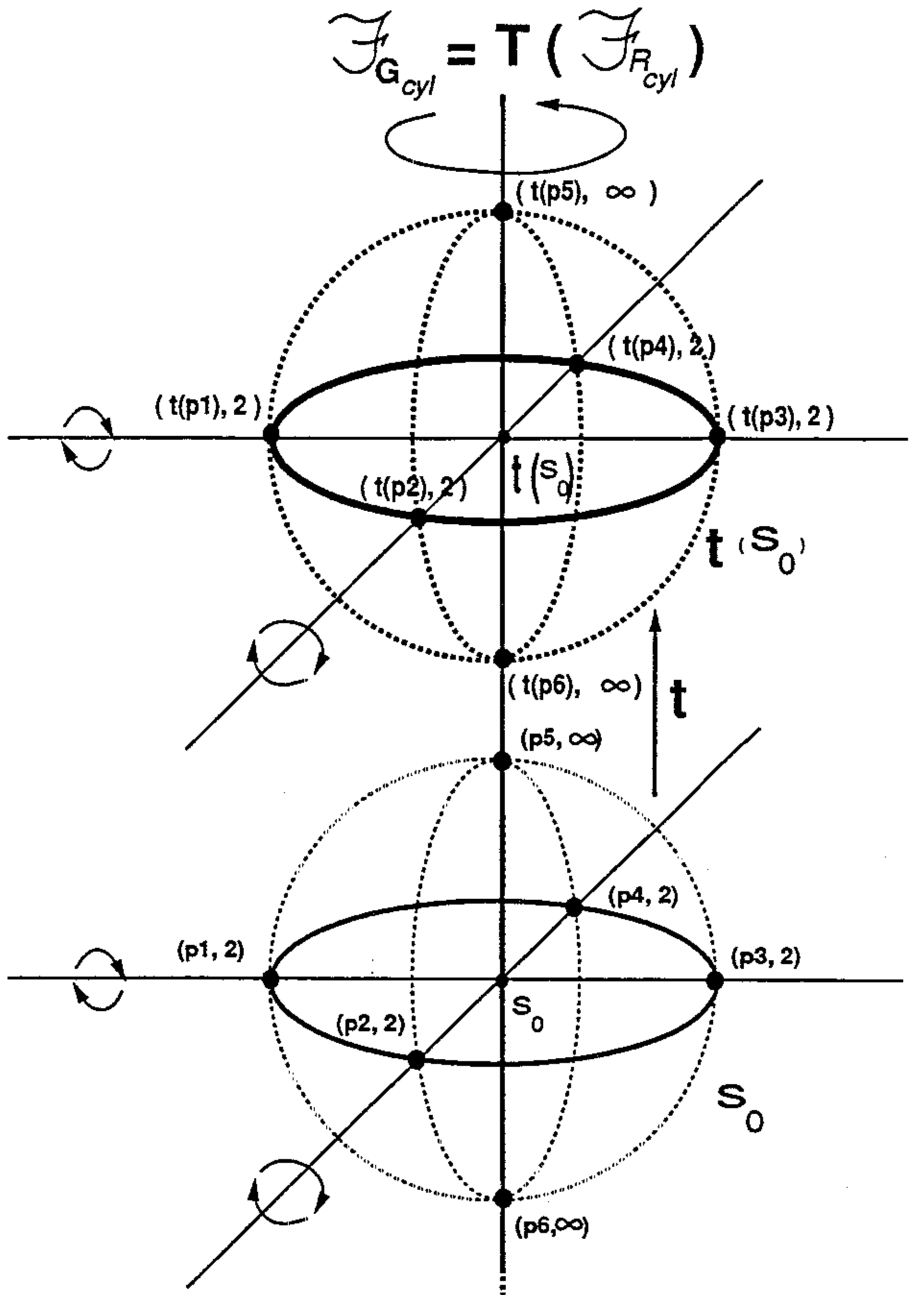

Figure 4.12: Rotation symmetries for an infinite cylinder: S_0 can be moved along $T(\mathcal{F}_{R_{cyl}})$.

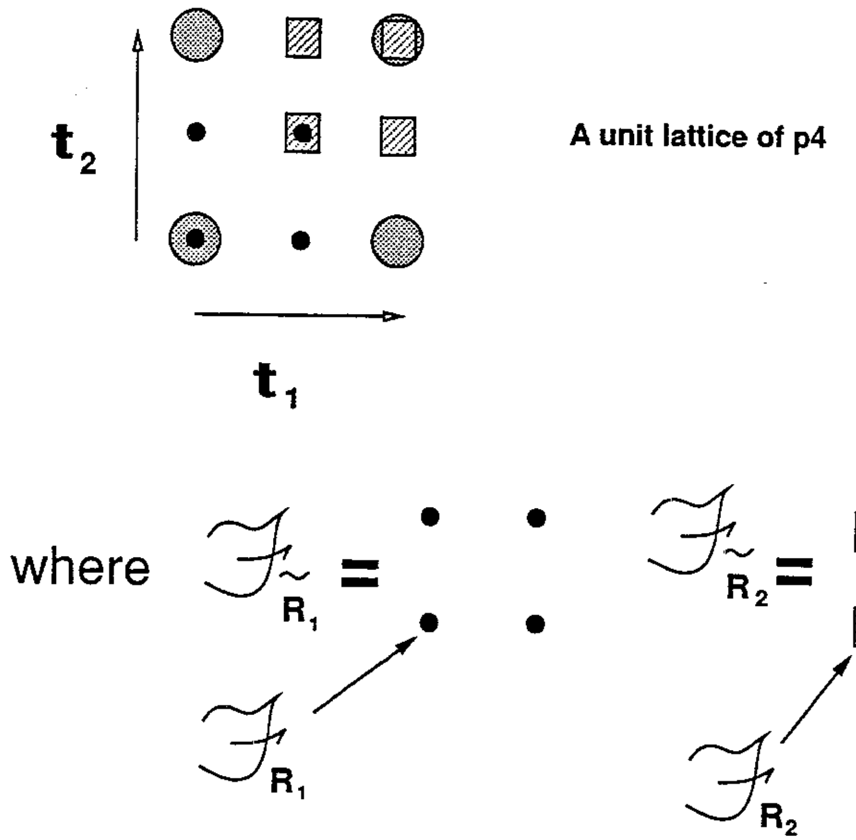

Figure 4.13: $G_1 = TR_1 = TR_2$. Since $\mathcal{F}_{R_1}$ and $\mathcal{F}_{R_2}$ each denotes a set of 4 points (lines), from $T(\mathcal{F}_{R_1}) = T(\mathcal{F}_{R_2})$ it is not necessarily true that there exists $t \in T$ such that $\mathcal{F}_{R_1} = t(\mathcal{F}_{R_2})$. Here t_1, t_2 are generators of T.

When T is a discrete subgroup of $\mathbf{T}^3$, however, it is not necessarily true that there exists $t \in T$ such that $\mathcal{F}_{R_1} = t(\mathcal{F}_{R_2})$ (Figure 4.13). Since $\mathcal{F}_{R_1} = tt'\mathcal{F}_{R_2}$ where $t \in T, t'$ is defined in expression 4.2, $R_1 = tt'R_2(tt')^{-1}$ (Proposition 4.50). By Lemma 4.68 and Corollary 4.58, $G_1 = TR_1 = Tt(t'R_2t'^{-1})t^{-1} = Tt'R_2t'^{-1} = TR_2 = G_2$ □

This Theorem justifies that there exists a bijection $\beta : \{G\} \rightarrow \{\mathcal{T}_G, (\mathcal{F}_G, \mathcal{P}_G)\}$

Definition 4.72 *Let* G_1, G_2 *be* TR *groups. The* **distance of TR groups** *is defined as*

$$\mathrm{dist}(G_1, G_2) = \mathrm{dist}(\mathcal{F}_{G_1}, \mathcal{F}_{G_2}) = \bigwedge_{s_1 \in \mathcal{F}_{G_1}, s_2 \in \mathcal{F}_{G_2}} \mathrm{dist}(s_1, s_2).$$

Lemma 4.73 *If* $G_1 = T_1R_1, G_2 = T_2R_2$ *are* TR *groups then there exist* $t_1 \in T_1, t_2 \in T_2$ *such that* $\mathrm{dist}(G_1, G_2) = \mathrm{dist}(\mathcal{F}_{t_1R_1t_1^{-1}}, \mathcal{F}_{t_2R_2t_2^{-1}})$.

Proof:

From Definition 4.72 $\mathbf{dist}(G_1, G_2) = \mathbf{dist}(T_1(\mathcal{F}_{R_1}), T_2(\mathcal{F}_{R_2}))$. Let $\mathbf{dist}(G_1, G_2) = \mathbf{dist}(s_1, s_2)$ where $s_1 \in T_1(\mathcal{F}_{R_1})$ and $s_2 \in T_2(\mathcal{F}_{R_2})$. There must exist $t_1 \in T_1, t_2 \in T_2$ such that $s_1 \in t_1(\mathcal{F}_{R_1}) = \mathcal{F}_{t_1 R_1 t_1^{-1}} \subseteq T_1(\mathcal{F}_{R_1}), s_2 \in t_2(\mathcal{F}_{R_2}) = \mathcal{F}_{t_2 R_2 t_2^{-1}} \subseteq T_2(\mathcal{F}_{R_2})$. Therefore $\mathbf{dist}(\mathcal{F}_{t_1 R_1 t_1^{-1}}, \mathcal{F}_{t_2 R_2 t_2^{-1}}) = \mathbf{dist}(G_1, G_2)$ (Lemma 4.13). □

4.4.4 TR Group Intersection

Theorem 4.2 *If* $G_1 = T_1 R_1, G_2 = T_2 R_2$ *and* $\mathbf{dist}(G_1, G_2) = \mathbf{dist}(R_1, R_2)$ *(By Lemma 4.73 such* R_1, R_2 *exist) then* $G_1 \cap G_2 = \beta^{-1}(\mathcal{T}_{T_1 \cap T_2}, (\mathcal{F}_{R_1 \cap R_2}, \mathcal{P}_{R_1 \cap R_2}))$.

Proof:

It is obvious that every $g \in (T_1 \cap T_2)(R_1 \cap R_2)$ is a member of $G_1 \cap G_2$. So we have $(T_1 \cap T_2)(R_1 \cap R_2) \subseteq G_1 \cap G_2$.

Now we prove the other inclusion $G_1 \cap G_2 \subseteq (T_1 \cap T_2)(R_1 \cap R_2)$:

A nontrivial member $g = tr$ of $G_1 \cap G_2$ can be one of the three kinds of isometries: a translation, a rotation or a screw motion, i.e. $t \| \bar{r}$.

If $g \in G_1 \cap G_2$ is a translation then $g \in T_1$ and $g \in T_2$ so $g \in T_1 \cap T_2$.

If g is a rotation then there exist $t_1 \in T_1, t_2 \in T_2$ such that $g \in t_1 \tilde{R}_1 t_1^{-1} \cap t_2 \tilde{R}_2 t_2^{-1}$ (Lemma 4.66 and Lemma 4.64). If one of T_1, T_2 is $\{1\}$ then $g \in R_1 \cap R_2$.

When T_1 and T_2 are both continuous, $g \in t_1 R_1 t_1^{-1} \cap t_2 R_2 t_2^{-1}$ (Lemma 4.64). Then for some $r_1 \in R_1, r_2 \in R_2, g = t_1 r_1 t_1^{-1} = t_2 r_2 t_2^{-1}$. By Proposition 4.31 $\bar{g} \| \bar{r}_1 \| \bar{r}_2$ and $\theta_g = \theta_{r_1} = \theta_{r_2}$.

If $\mathbf{dist}(G_1, G_2) = 0$ then $\mathbf{dist}(R_1, R_2) = 0 \Rightarrow \mathcal{F}_{R_1} \cap \mathcal{F}_{R_2} \neq \emptyset \Rightarrow \bar{r}_1 \cap \bar{r}_2 \neq \emptyset$ Thus $\bar{r}_1 = \bar{r}_2 \Rightarrow r_1 = r_2 \in R_1 \cap R_2$, and so $t_1 = t_2 = t \in T_1 \cap T_2$. Therefore $g \in t R_1 t^{-1} \cap t R_2 t^{-1} = t(R_1 \cap R_2) t^{-1}$.

If $\mathbf{dist}(G_1, G_2) > 0$ i.e. $T_1(\mathcal{F}_{R_1}) \cap T_2(\mathcal{F}_{R_2}) = \emptyset$ then $\mathbf{dist}(R_1, R_2) > 0$. Obviously, neither of T_1, T_2 is isomorphic to $\mathbf{T}^3$ so $\mathcal{T}_G$ is either a line or a plane. In order to have a nontrivial rotation in $t_1 R_1 t_1^{-1} \cap t_2 R_2 t_2^{-1}$, $t_1 R_1 t_1^{-1}$ and $t_2 R_2 t_2^{-1}$ both have to be point groups (recall the proof in Proposition 4.53). Thus $t_1(\mathcal{F}_{R_1}) \in \bar{g}, t_2(\mathcal{F}_{R_2}) \in \bar{g}$. By the **TR**-restriction, if $\bar{g} \| T_1$ then $\bar{g} \in T_1(\mathcal{F}_{R_1}) \Rightarrow t_2(\mathcal{F}_{R_2}) \in T_1(\mathcal{F}_{R_1}) \Rightarrow T_1(\mathcal{F}_{R_1}) \cap T_2(\mathcal{F}_{R_2}) \neq \emptyset$, a contradiction. Similarly, it can be shown for $\bar{g} \| T_2$. If $\bar{g} \perp T_1$ and $\bar{g} \perp T_2$ then $\bar{g} \perp T_1(\mathcal{F}_{R_1})$, $\bar{g} \perp T_2(\mathcal{F}_{R_2})$ and $\bar{g} \cap T_1(\mathcal{F}_{R_1}) = t_1(\mathcal{F}_{R_1})$, $\bar{g} \cap T_2(\mathcal{F}_{R_2}) = t_2(\mathcal{F}_{R_2})$ (Figure 4.14). This implies $\mathbf{dist}(T_1(\mathcal{F}_{R_1}), T_2(\mathcal{F}_{R_2})) = \mathbf{dist}(t_1(\mathcal{F}_{R_1}), t_2(\mathcal{F}_{R_2})) = \mathbf{dist}(G_1, G_2) = \mathbf{dist}(\mathcal{F}_{R_1}, \mathcal{F}_{R_2})$ (Definition 4.12) (Figure 4.14). It follows then $t_1 = t_2 = t \in T_1 \cap T_2 \Rightarrow g \in (T_1 \cap T_2) t (R_1 \cap R_2) t^{-1}$.

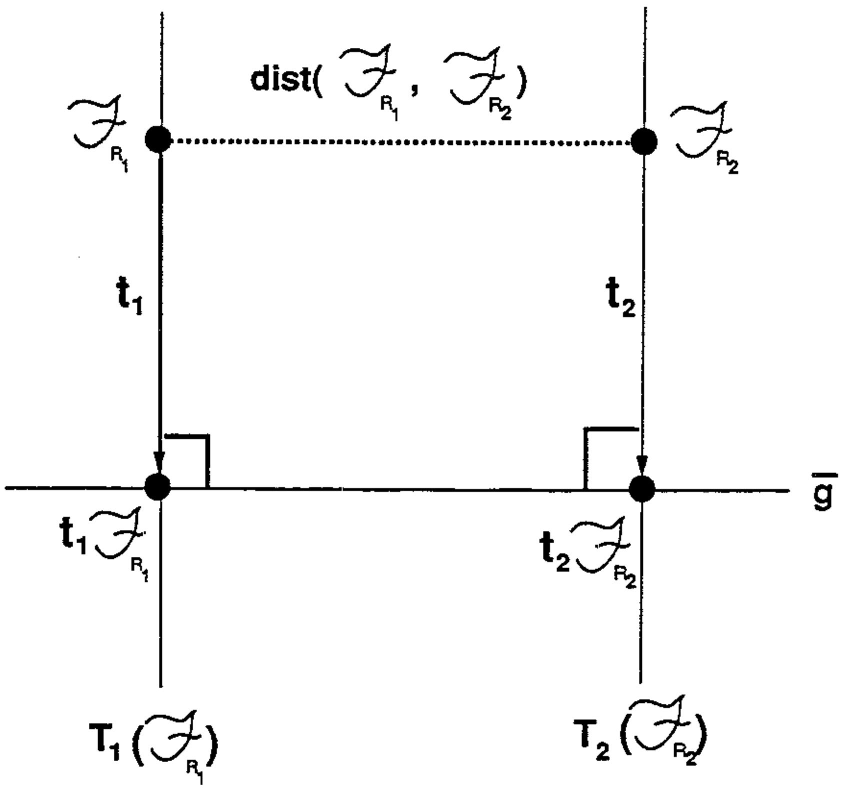

Figure 4.14: When R_1, R_2 are both point groups and $\bar{g} \perp \mathcal{T}_G$

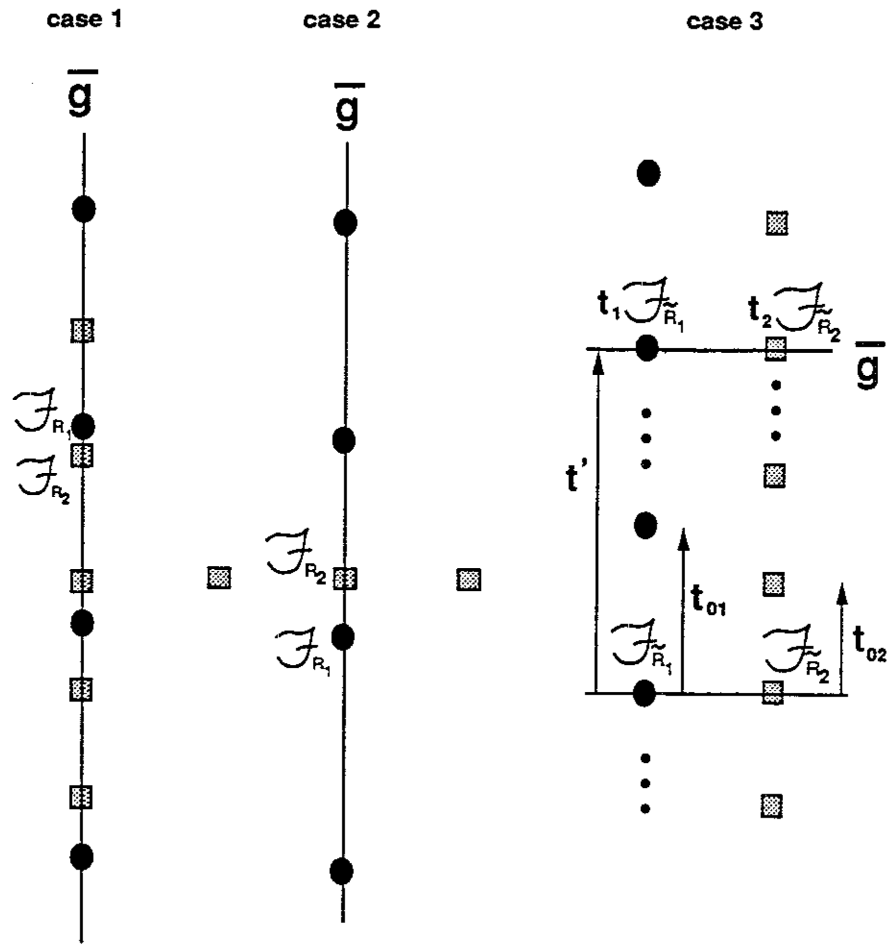

Figure 4.15: When T_1, T_2 are both one dimensional discrete groups.

From $g \in (T_1 \cap T_2)t(R_1 \cap R_2)t^{-1}$, it follows that $G_1 \cap G_2 \subseteq (T_1 \cap T_2)t(R_1 \cap R_2)t^{-1} \Rightarrow G_1 \cap G_2 = (T_1 \cap T_2)t(R_1 \cap R_2)t^{-1}$. By Corollary 4.58 $G_1 \cap G_2 = (T_1 \cap T_2)(R_1 \cap R_2)$.

When one of T_1, T_2 is discrete, a proof to the above can be given. The only difference is that a single rotation group R_1 or R_2 is replaced by a set of rotation groups in the generating region, namely $\tilde{R}_1, \tilde{R}_2$.

When both T_1, T_2 are discrete, for instance when both are one dimensional discrete groups, then $\mathcal{T}_{G_1}, \mathcal{T}_{G_2}$ are both one dimensional lattices. Due to the TR-restriction, there are only three possible cases for g to be a nontrivial rotation: $\bar{g} \| \mathcal{T}_{G_1}$ and $\bar{g} \| \mathcal{T}_{G_2}$; $\bar{g} \| \mathcal{T}_{G_1}$ but $\bar{g} \perp \mathcal{T}_{G_2}$ (or vice versa); and $\bar{g} \perp \mathcal{T}_{G_1}$ and $\bar{g} \perp \mathcal{T}_{G_2}$, as shown in Figure 4.15. Since $\mathbf{dist}(R_1, R_2) = \mathbf{dist}(G_1, G_2)$,

- case 1 and case 2: it has to be true that $g \in R_1 \cap R_2$.

- case 3: since $\mathbf{dist}(t_1R_1t_1^{-1}, t_2R_2t_2^{-1}) = \mathbf{dist}(R_1, R_2)$, either $t_1 = t_2 = 1$ or $t_1 = t_2 = t^n = t'$. Here n is an integer and $t = lcm(t_{01}, t_{02})$, the least common multiple for the norms of the generators t_{01}, t_{02} of T_1 and T_2 respectively. Therefore $g \in t'(R_1 \cap R_2)t'^{-1} \Rightarrow g \in (T_1 \cap T_2)(R_1 \cap R_2)$.

The basic arguments are the same while the dimensions of T_1, T_2 increase.

If g is a screw motion then $g = t_1r_1 = t_2r_2$ where $t_1 \in T_1, r_1 \in R_1, t_2 \in T_2, r_2 \in R_2$ and $\bar{r}_1 \| t_1, \bar{r}_2 \| t_2$. Then we have $t_2^{-1}t_1 = r_2r_1^{-1} \in \mathbf{T}^3$. Since $t_2^{-1}t_1$ is a translation perpendicular to the translation $r_2r_1^{-1}$, for two such translations to be equal is to let both be 1. Therefore $t_2^{-1}t_1 = 1 \Rightarrow t_1 = t_2$ and $r_2r_1^{-1} = 1 \Rightarrow r_1 = r_2$. Therefore $g \in (T_1 \cap T_2)(R_1 \cap R_2) \Rightarrow G_1 \cap G_2 \subseteq (T_1 \cap T_2)(R_1 \cap R_2)$.

Thus $G_1 \cap G_2 = (T_1 \cap T_2)(R_1 \cap R_2)$. Clearly, $\beta^{-1}(\mathcal{T}_{T_1 \cap T_2}, (\mathcal{F}_{R_1 \cap R_2}, \mathcal{P}_{R_1 \cap R_2})) = (T_1 \cap T_2)(R_1 \cap R_2)$ (Theorem 4.1). Therefore $G_1 \cap G_2 = \beta^{-1}(\mathcal{T}_{T_1 \cap T_2}, (\mathcal{F}_{R_1 \cap R_2}, \mathcal{P}_{R_1 \cap R_2}))$.
□

4.5 An Algorithm Outline for TR Group Intersection

Based on the Theorems in the previous section, we have implemented a TR group intersection algorithm using characteristic invariants[6]. The diagram of the algorithm is shown in Figure 4.16. The current implemented version[7] has asymptotic time complexity $\mathcal{O}(n^2) + \mathcal{O}_B(b \log^2 b \log \log b)$ where n is the number of poles, and $\mathcal{O}_B(b \log^2 b \log \log b)$ is the order of magnitude for computing the greatest common divisor under the bitwise computation model. The only case where n can be arbitrarily large is when one of the groups is a dihedral group. However this case can be taken care of by using generator representations for the poles of dihedral groups.

Figure 4.16 illustrates the outline of the TR group intersection algorithm. Note, the subcases for computing the fixed point set $\mathcal{F}$ is not listed in Figure 4.16 but can be found in great detail in Proposition 4.53.

Propositions 4.35 and 4.37 establish the justification for canonical representation of each TR groups, which has been used in the implementation of the algorithm.

[6] Actually, the algorithm was implemented long before the proofs were done.

[7] This version does not yet include the Crystallographic groups. For an algorithm outline dealing with the intersection of Crystallographic groups see Appendix D.

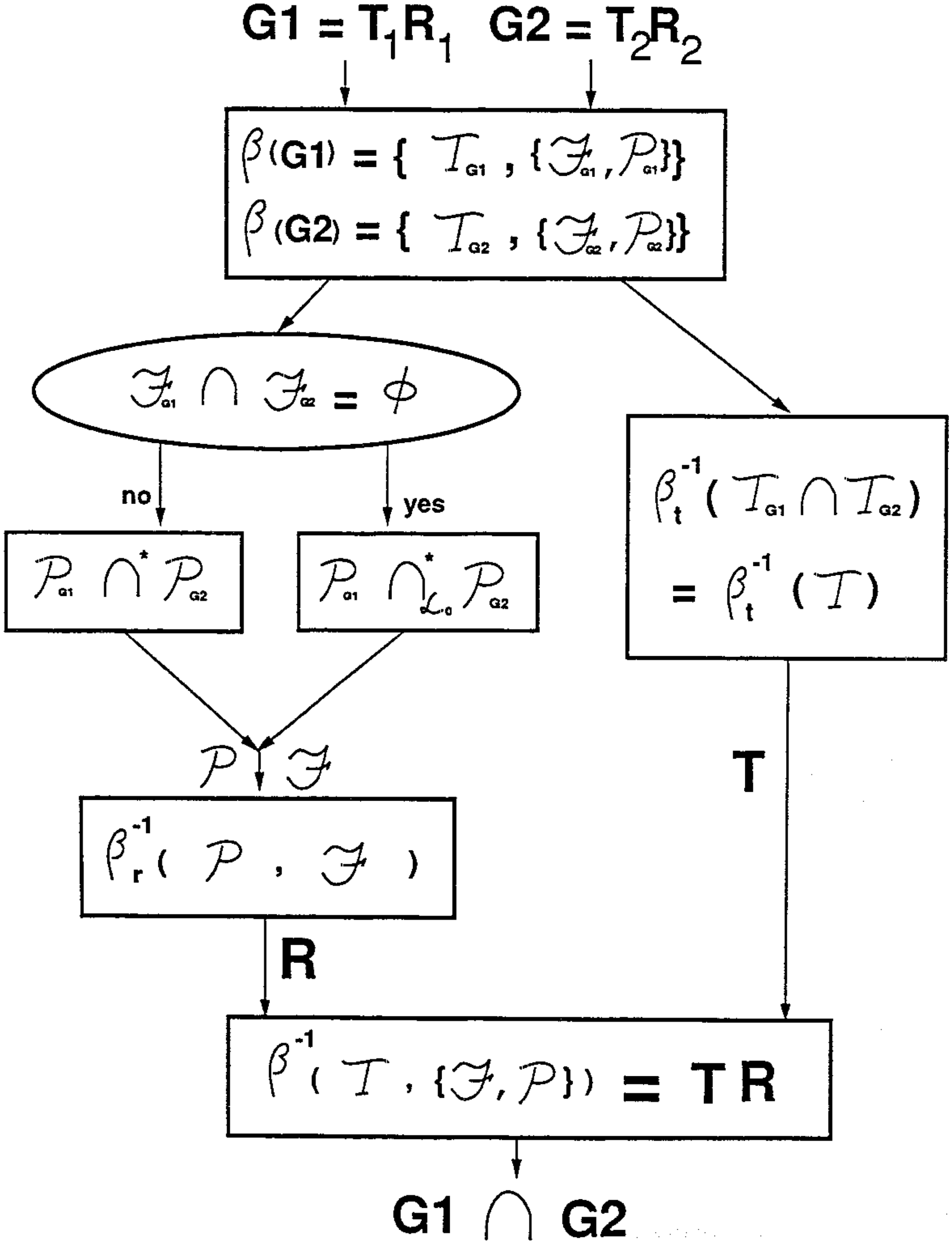

Figure 4.16: The outline of the TR group intersection algorithm

4.6 Summary

In this chapter we have studied a particular family of subgroups of the proper Euclidean group $\mathcal{E}^+$, namely the **TR** subgroups. The main results include:

a) We defined **TR** subgroups of $\mathcal{E}^+$ as $G = TR$ where T is a translation subgroup and R is a rotation subgroup. The relative independence of the translations and rotations of the **TR** group gives us the freedom to characterise translations and rotations separately. The only symmetry group we are interested that cannot be represented as a **TR** group is the symmetry group of a screw, in which translations are not independent from rotations (see $\mathcal{G}_{screw}$ in Table 3.1). We have, however, included the screw group as a 'black sheep' of the family in our planning system.

b) We defined characteristic invariants for **TR** groups and proved their one-to-one correspondence. This brings the geometric aspect of the **TR** groups to the foreground for computational purposes, while still preserving their algebraic nature.

c) We proved that the intersection of **TR** groups can be obtained from the calculation of their characteristic invariants, thus justifying an efficient **TR** group intersection algorithm.

CHAPTER 5

AN ASSEMBLY PLANNING SYSTEM $\mathcal{KA}3$

"The first thing I've got to do", said Alice to herself, as she wandered about in the wood, "is to grow to my right size again; and the second thing is to find my way into that lovely garden I think that will be the best plan."

It sounded an excellent plan, no doubt, and very neatly and simply arranged: the only difficulty was, that she had not the smallest idea how to set about it ....

Lewis Carroll, **Alice's Adventures in Wonderland**

Each assembly part treated in this work is assumed to be a three dimensional *rigid body*[1]. Figure 5.1 gives an overview of the assembly planning system $\mathcal{KA}3$. There are two kinds of inputs to $\mathcal{KA}3$:

- complete knowledge of the *nominal* geometry and topology of each body involved in the assembly, provided by the geometric solid modeller PADL2 [11] (Appendix E).

- kinematic constraints provided by the user. A *constraint* is a required relationship among one, two or more entities. Some typical constraints are shown in Table 5.1. Users can specify the relationships among bodies only without mentioning how exactly body features are related. Constraints upon the order of the assembly operations which establish these relationships can also be given.

From these inputs $\mathcal{KA}3$ proceeds in three stages:

- stage one: proposing possible assembly configurations;

- stage two: checking the feasibility of each assembly configuration;

- stage three: disassembling each feasible assembly configuration.

[1] A mathematical abstraction, defined as a continuum for which the distance between any pair of its points remains unchanged [2] under *any physically possible motion*

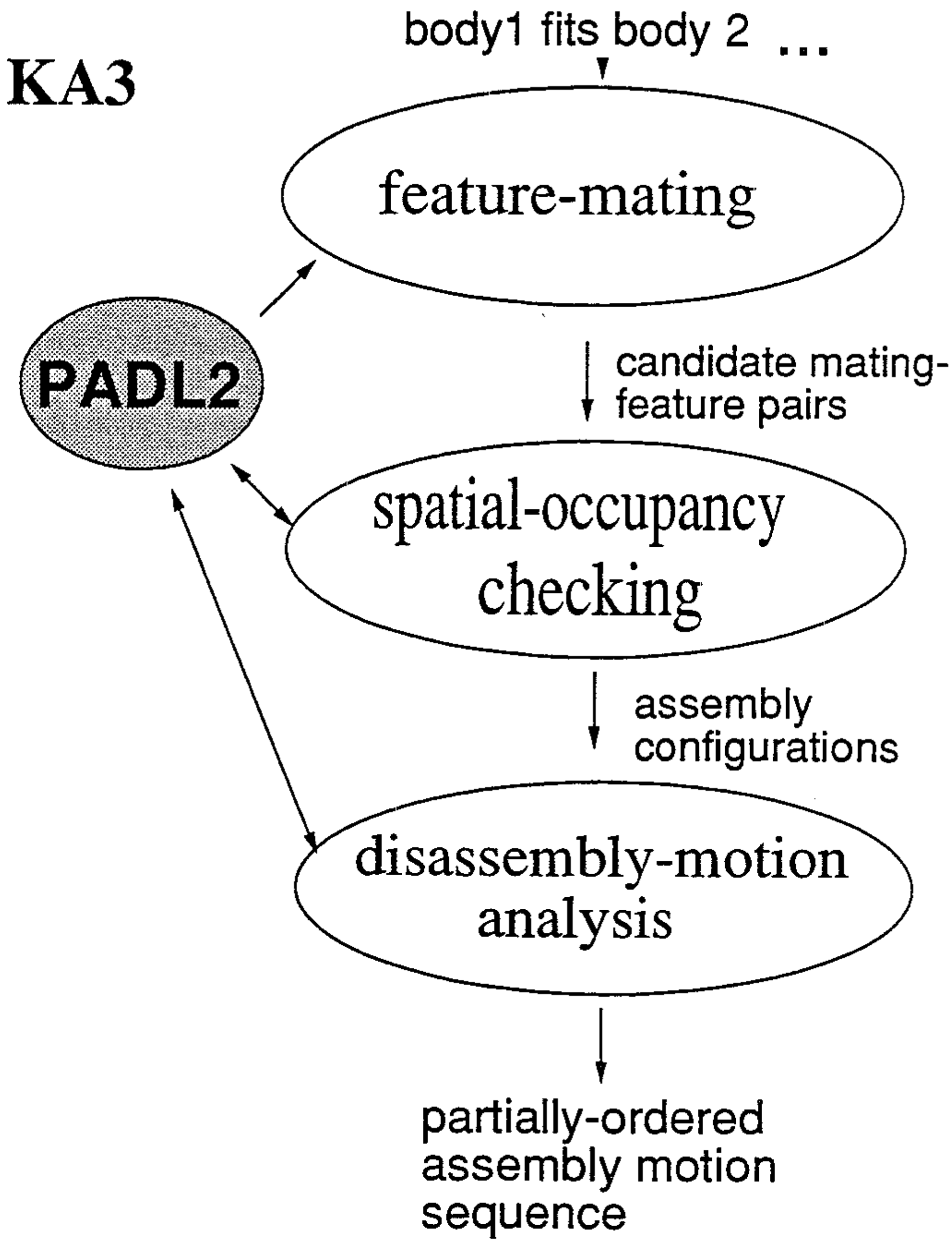

Figure 5.1: An overview of *KA3* system

Table 5.1: Some typical constraints given by users

Rel	entities	nature	definition
fit	(body1,body2) *or* (f1_of_b1,f2_of_b2)	kinematic	area contact
mesh	(body1,body2) *or* (gear1_of_b1,gear2_of_b2)	functional kinematic	teeth meshed into each other
before	(Rel1,Rel2)	temporal	establish Rel1 before Rel2

5.1 Assembly Feature-mating Inference from Solid Models Using Symmetry Groups

The problem to be solved in this stage is to find a set of candidate mating feature pairs from the boundary model of each body and the given kinematic constraints.

5.1.1 How to Describe Shapes to PADL2

We use Prolog terms to express Constructive Solid Geometry as described in [64]. This formalism is translated into the input syntax for PADL2 by the Prolog predicate *draw*. All of the involved assembly parts are treated thus. For bodies $b1$ and $b4$ in Figure 5.3, 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 3.2, page 29).

5.1.2 How PADL2 Represents Solids

The internal representation of the boundary model produced by PADL2 is extracted using FORTRAN subroutines linked into POPLOG [29, 78] 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
```

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. Table 3.2 shows some corresponding surface types and symmetry groups. The reason that infinitely extended surfaces and their symmetry groups in Table 3.2 are important and useful is that finite solids are bounded by such surfaces. We have discussed features and their symmetry groups in detail in Chapter 3, where a primitive feature is defined to be an algebraic surface (Definition 3.11, page 25), and a compound feature is the set union of primitive features (page 30).

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 recordclasses (only
                            ;;;  one for a primitive feature)
```

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.

5.1.3 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, its symmetry group is computed by intersecting the symmetry groups of its primitive features.

Compound Feature Recognition

The feature library was built based on a survey of some of the typical features in assembly [61] 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. See Appendix F for a precise definition for insertor and container type features. 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, $B_1, B_2, ..., B_n$ which have $m_1, m_2, ..., m_n$ faces 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 a body 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 surfaces and surfaces with texture such as *threads* or *gears*, have higher priority to be matched first. Because such surfaces with given parameters are relatively rare, 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.

Compound Feature Computation

Let us recall that a *compound feature* F_{comp} of body B is a set of primitive features F_i of B. Provided the F_is are all distinct (Definition 3.15, page 25; Proposition 3.18, page 30) the symmetry group of F_{comp} is

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

The algorithm developed in Section 4.5 has been applied to compute $sym\_group(F_{comp})$. In cases where $F_{comp} = F_1 \cup F_2$ and F_1 and F_2 are *strongly congruent* then the symmetry group of F_{comp} is the product of the above group intersection and a conjugated subgroup of $SO(3)$ (Proposition 3.20, page 32). For example, one *multi-insertor* compound feature of $b1$ in Figure 5.3, 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 5.2).

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

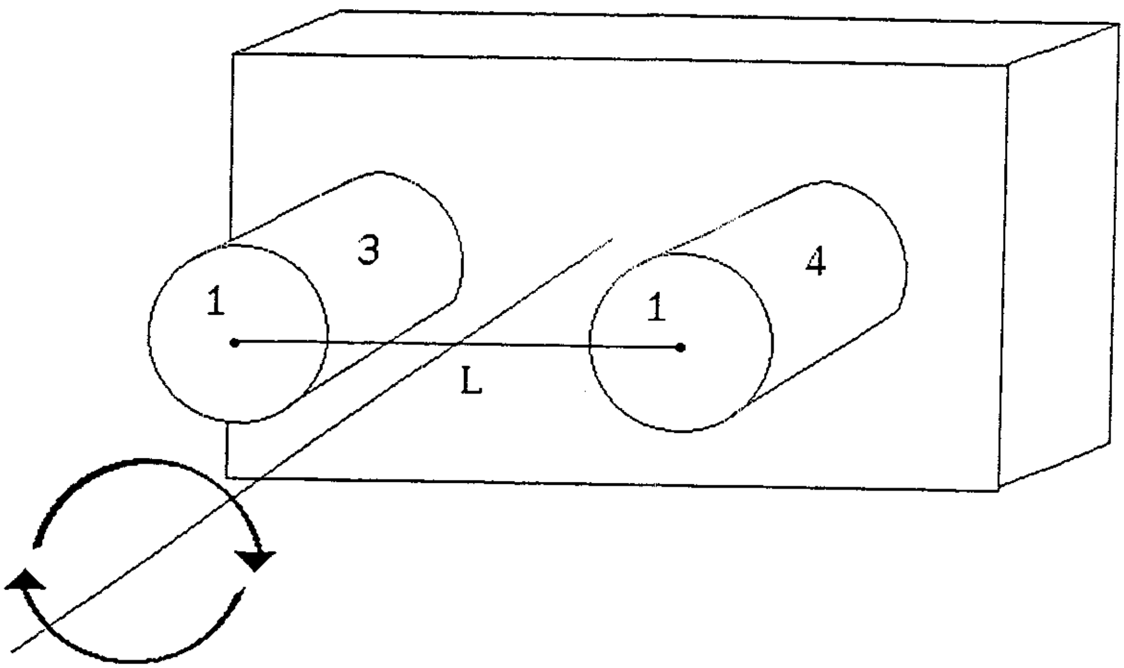

Figure 5.2: Extra symmetry in a compound feature

choice of Z axis should be the line parallel to the center lines of the two cylinders and midway between them (Figure 5.2).

5.1.4 How to Find Mating Features

Since the size of the solution space for mating features is very large, we try to identify certain necessary conditions for two features to mate. Any candidate mating feature pair which does not satisfy the necessary conditions can be excluded immediately from further searching.

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. As we have pointed out in Chapter 3 fitting features have the same symmetry group. Thus the necessary condition 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 provided in the PADL2 boundary model. Two features to be mated 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 fit are:

- they have the same symmetry group,
- they have opposite concavity/convexity, and
- their dimensions are consistent.

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 the 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 has been discussed in Chapter 3 and will be used in the next stage to satisfy both kinematic and spatial constraints.

5.2 Assembly Planning via Kinematic and Spatial Constraint Satisfaction

A constraint satisfaction problem (*CSP*) is defined over finite discrete domains in [56] as:

> Given a set of n variables each with an associated domain and a set of constraining relations each involving a subset of the variables, find all possible n-tuples such that each n-tuple is an instantiation of the n variables satisfying the relations. Each of such n-tuples is a *solution*.

The problem we are facing in this stage is this:

Given : a set of kinematic constraints, each of which is associated with a set of mating feature pairs that satisfy the necessary mating feature conditions described in Chapter 5.1 [49, 50],

Find: all the possible assembly configurations in terms of mating features between bodies and relative positions of bodies, *Satisfy*: the kinematic and spatial constraints along with an optimality measure: maximizing the number of contacting surfaces.

We shall call this the *assembly constraint satisfaction problem* (ACSP) to distinguish it from the general CSP. Each solution of ACSP is called a *feasible assembly configuration*. One interesting aspect of this ACSP is that one of the constraints to be satisfied is also a variable. Here, each given kinematic constraint, associated with a set of candidate mating feature pairs, is actually a variable in ACSP.

In Section 5.1 we describe in detail how to find candidate mating features from a set of geometric boundary models of bodies [11], by matching against a salient-feature library. Now let us assume that such mating feature pairs have been found, as have the relationships between each pair of mating features. These relationships are described in terms of symmetry groups and are called the *symmetry constraints* The possible mating features form domains of finite sizes, thus ACSP appears to be a CSP over finite sets. However a set of candidate mating feature pairs, defining an assembly configuration, also has to satisfy the spatial constraint to determine which we have to know the relative positions of bodies subject to the symmetry constraints. This is an embedded CSP over possibly infinite domains.

Besides the interleaving of different constraints, *KA3* differs from the SPAR system [38] and Wolter's XAP/1 [84] in that a constraint between bodies is expressed as the *relative location* of the bodies belonging to a (possibly infinite) generalized coset of feature symmetry groups. This relative notion is more expressive and relevant for assembly operations.

It is known that, in a general *CSP*, finding one consistent solution is an NP-complete problem. In ACSP, clues deduced from the symmetries of the assembly are used to reduce the sizes of the variable domains, thus mitigating the combinatorics, as described in the next section.

5.2.1 Symmetry Constraints and Constraint Satisfaction Process

In this section we describe in principle how the ACSP is treated. As we pointed out at the beginning of this chapter, the given requirements among bodies of an assembly are treated as variables in *ACSP* and the possible mating features form the domain for each variable. If the number of variables is N, the domain size is D (average), and the number of constraints is C, then the search space would be on the order of $O(CD^N)$. This shows the importance of reducing the domain size D.

A three-step-algorithm for finding a solution in ACSP follows:

Step one is setting up the constraint satisfaction graph. Given the non-instantiated kinematic constraints (input) and all the possible mating features, the system establishes a two-layered graph. The top layer has bodies as nodes and required relationships among bodies as arcs (this is a dual graph of the actual constraint satisfaction graph). The bottom layer has candidate mating features as nodes and the common symmetry groups between mating features as arcs. Whenever the domain size of one of the variables is zero the process terminates. Whenever two candidate mating features have the *identity* group as their common symmetry group, or the common symmetry group is the same as the symmetry group of one of the bodies on which the features reside, the relative position of the two related bodies can be uniquely determined and the *intersect* predicate of PADL2[2]can then be called for spatial occupancy checking. In this case, node-consistency is achieved. Node-consistency can be checked only when the common symmetry group is

- the identity group, or
- the same as its body symmetry group, or
- a finite symmetry group (it this case, all the possibilities must be checked).

Step two is reducing the domain size of each variable. The following set of rules are applied in turn until the network becomes quiescent:

- **Mandatory request law**: if there is *only* one correspondence at the bottom level with an upper layer input constraint this mating feature pair has the highest priority. Therefore if one of the features is a non-**sharable** feature[3], then any other arcs pointing to this feature should be deleted. The precondition of this rule corresponds to the case in *CSP* where the domain size of a variable is one. The effect of applying this rule can be propagated until the network is quiescent.

  The more interesting consequence of this law is its extension to multiple variables over domain size larger than one. This happens when there are n variables demanding values (non-sharable ones) from the same domain D with

[2]PADL2's *intersect* predicate returns *true* if no interference of two specified bodies is found, *false* otherwise.

[3]A predicate called **sharable** is applicable to a pair of features from distinct bodies. It returns *true* if a third feature can also be mated simultaneously, *false* otherwise.

size n. When this happens, these n variables form a *cluster* of size n which completely take over domain D. Thus any other variable v_i outside of this cluster, with domain D_i should reduce its domain to $D_i - D$. If this causes any domain to become empty, the process terminates.

Besides the *sharable* predicate among features of different bodies, another relation *co-existing* among features residing on the same body should also be established. This means that if f_1, f_2 are non-co-existing, when f_1 is chosen f_2 loses its status of an eligible compound feature.

- **Equivalent feature law.** A pair of features of a body is said to be *equivalent* if they have the same symmetry group, the same dimensions and belong to the same *orbit*.

  A set of features $F_1, F_2, \ldots F_n$ of body B belong to the same *orbit* if and only if there exists $g \in G$ (the symmetry group of the body) such that $g(F_i) = F_{i+1}, g(F_n) = F_1$. Since $g(B) = B$, the features permute among themselves under g while the body sits still set-wise. Being able to represent and reason about these characteristics, the system will choose only one out of a set of equivalent features. The reasoning for the chosen one can be applied to the rest of the features in its orbit. For example, consider a finite cylinder which has two identical ends: all the reasoning about one end of the cylinder applies to the other end of the cylinder. This rule shall be applied unless it causes the termination of the process.

- **Arc/Path consistency checking** [56, 59] is a local consistency checking method for CSPs which can be used to eliminate those arcs which will not lead to a conclusion.

Step three is checking the spatial constraint globally. Firstly, we need to find all the subgraphs in the bottom level (relating mating features) which are isomorphic to the top level graph (relating bodies). Each of these subgraphs is an assembly configuration. Secondly, the spatial constraint is applied. When the relative position of a pair of related bodies cannot be determined locally in step one, this is the time to evaluate the situation globally, i.e. to see whether constraints from other bodies can be used to decide the position. We have experimented with a simple and quick algorithm due to Kramer [42] that solves certain algebraic equations geometrically. This will be explained further in the examples.

Complexity analysis of the above algorithm:

Step one takes $O(nd)$ time where n is the number of relationships among bodies and d is the domain size.

Step two uses the arc/path consistency checking algorithms described in [59] which has complexity $O(nd^2)$ where d is the bound for domain sizes.

Step three needs to check $s_1 \times s_2 \ldots \times s_n$ combinations where n is still the number of kinematic constraints and s_i is the *reduced* domain size.

In summary, this stage starts by establishing the two layered constraint satisfaction graph. Whenever possible, node consistency checking is carried out. The feature-level graph is pruned by applying rules in step two including arc/path consistency checks. Finally, isomorphic subgraphs are mapped from the feature-level to the body-level where the spatial occupancy constraint is applied to each. The original network is thus transformed into smaller and smaller networks such that the final isomorphic graph matching can be performed on a much slimmer graph.

5.3 Disassembly-motion Analysis

The output of stage two is a set of graphs that satisfy both the kinematic and the spatial constraints. Each of the graphs denotes a feasible assembly configuration (FAC). Each FAC contains information about how the bodies are related in terms of symmetry groups of the corresponding mating features. The problem to be solved in this stage is finding whether a FAC can be disassembled via translational motions in order to find how to assemble the FAC; this is plausible because the process of assembly/disassembly is reversible if the components are rigid and there is no internal energy stored[86]. Much of the following terminology is borrowed from Woo's paper on automatic disassembly and total ordering in three dimensions [86]:

Definition 5.1 *A body is* **cleared** *from a subassembly if the body can be translated to infinity.*

Definition 5.2 *A component is* **m-disassemblable** *if m motions are needed to clear it.*

Definition 5.3 *Suppose an assembly Σ has n components. We say Σ is* **m-separable** *if m is the smallest number of bodies which have to be taken off Σ* simultaneously *to ensure a collision-free disassembly path.*

Corollary 5.4 *If an assembly Σ is m-separable then either $m < n$ or $m = n$. In the second case if $m = n > 1$ we call Σ non-disassemblable.*

Definition 5.5 Σ *is disassemblable iff* Σ *has only one component or none of its subassemblies is non-disassemblable. (This is a recursive definition)*

At this stage we are only concerned with 1-separable assemblies. In each case of subassembly, only 1-disassemblable bodies are searched for and we also restrict the motion to clear the body to be a translational motion. If no such body exists then no solution will be produced. However, when no single body in an assembly can be translated to infinity without collision, it does not necessarily imply there is no way to disassemble the assembly. This restriction is not as severe as it may seem. Most assemblies are 1-separable.

There are three subtasks to be accomplished :

1) to find which body or bodies of the given FAC are movable by a translation;

2) to find whether there is any spatial interference along the translation trajectory (to infinity) of such a body;

3) to record the contact states as a partially ordered assembly motion sequence (POAMS).

A *mechanism* is an assembly of rigid bodies called *links* that are connected together at *joints*. An assembly is a coupling of rigid bodies which constitutes a *kinematic chain*. A *joint face pair* (JFP) is a pair of faces on two separate links that overlap, have opposing normals at all points of overlap [39]. An assembly configuration thus is a set of JFPs. This is a helpful viewpoint in which a robot hand grasping an object can be considered as assembling the face of the hand to the object to form a JFP.

5.3.1 Degree of Freedom Analysis

The degree of freedom (d.o.f.) between two coupled links is defined as the number of independent variables required to define the relative position of the two coupled links. There are two different measurements depending on the conditions associated with the above definition, these two conditions being:

- the coupled links maintain contact at the next instance of time — which is called *instantaneous kinematics*, and

- the coupled links can be separated at the next instance of time — all possible motions are allowed. This is what we are concerned at this stage.

For example, a peg in a non-through hole has an $SO(2)$ group associated with the joint between the peg and the hole, from Table 2.1 (page 19) $dim(SO(2)) = 1$, and the only motion possible for the peg is rotation about its central axis. When the peg is lifted, the bottom surfaces are separated and the mating feature symmetry group becomes $\mathcal{G}_{dir\_cyl}$ and $dim(\mathcal{G}_{dir\_cyl}) = 2$, The peg now can move along its central axis as well as rotate about it. The relation just broken gives a bound on the motion between the peg and the hole.

A systematic way of breaking compound feature relationships is achieved by combining the symmetry group representation with Japokac's possible motion versus constrained motion list. The basic idea is:

- Form a table of canonical groups associated with a list of possible and constrained motions, for example, for the planar and cylindrical lower joints see Table 5.2.

- Each body in an assembly can be considered as being related to the rest of the assembly by one compound feature, this compound feature is composed of a set of primitive features, i.e. JFPs. Each JFP has a symmetry group and each such group has a motion list, an entry from Table 5.2. When two JFPs are combined into a new JFP their motion lists will be combined as described below. To determine whether a body can move is to intersect the possible motions from all the JFPs of the body and find whether the final possible motion list is non-empty.

  Some motions will keep the specific relation of a JFP, others will break the relation. These two kinds of motions are listed under different columns in Table 5.2 where **k** is a vector in the direction.

- Using the constrained motions, delete some motions of the body, leaving the possible motions the body can take. The complement of the resulting possible motions is the final constrained motion. Note, the same symmetry group may have a different motion list associated with it depending upon the specific geometry of its JFPs.

5.3.2 Spatial Interference Checking

As a result of d.o.f. analysis, one body or a set of bodies is identified as being movable. The body can then be moved away in the direction allowed by its kinematic

Table 5.2: Possible Motions versus Restricted Motions

Contacting Feature Symmetry Group	Possible Motions		Restricted Translational Motions
	Maintain contact	Break contact	
$\mathcal{G}_{plane}$	$\vec{v} \cdot \mathbf{k} = 0$	$\vec{v} \cdot \mathbf{k} > 0$	$\vec{v} \cdot \mathbf{k} < 0$
$\mathcal{G}_{cylinder}$	$\vec{v} \parallel \mathbf{k}$		$\vec{v} \not\parallel \mathbf{k}$

constraints. One way to find whether there is a collision-free path is to sweep the body along the trajectory and intersect the swept volume with the rest of the assembly. However, PADL2 does not support the sweeping operation. A geometric modeller system such as the one described in [12] would be ideal. In $\mathcal{KA}3$ spatial interference checking which involves sweeping has been handled by calling special purpose functions.

By analyzing the relative motions within an assembly at least one component should be identified as movable, otherwise 'no solution' will be reported. If a path for moving away the designated body exists, the order in which a compound relationship is broken should be recorded. Such as moving a peg away from a blind-hole, the bottom plane-plane relation is broken first, then the cylindrical surfaces are separated. The disassembly in terms of the symmetry groups can thus be described as: $SO(2) \rightarrow \mathcal{G}_{dir\_cyl} \rightarrow \mathcal{E}^{+}$.

5.3.3 Disassembly Order Analysis

The above analysis is in preparation for finding the order of disassembling mated bodies which are in contact. Some ordering constraints may be already given by the user or occur during the reasoning of stage one [45]. Those constraints demand some decisions to be made about the order of a disassembly. The system follows the *least commitment* strategy, in which no decisions are made until mandated by some constraints. A *partially ordered sequence* will be formed because two 1-disassemblable bodies have no precedence relation to each other when no constraints are present.

After a body is moved away its node and related arcs will be deleted from the FAC network. Repeating this process until only one body is left gives us a series of states arising during the disassembly of the original configuration. Reversing the series, a *partially ordered assembly motion sequence* (POAMS) is found. The *critical positions*, which are those positions where one or more relationships are broken/established, should be identified so that the range of the motions is provided.

Sensor requests can be embedded in the final plan to verify the start/termination of a relation. The sequence of these states of subassemblies gives a *precedence diagram*(s) which can be represented as a directed AND/OR graph [35].

Each POAMS is represented as a directed graph with *contact states* as nodes and translational-motions as arcs. The data structure for contact states is defined as

```
recordclass contact_state ;;; a node on the contact state graph
        p_state_name   ;;; name of the contact state
        symmetry_gr    ;;; a subgroup of E, an instance of group
                       ;;; recordclass gr
        mating_feat    ;;; a list of pairs of mating features, each
                       ;;; of such pairs is expressed by a triple:
                       ;;;   [ operation feature1 feature2]
        p_transitions  ;;; a list of arcs to lead to other states
                       ;;; from the current one
;
```

and for each motion

```
recordclass preced_action  ;;;a translation in our current version
     paction_name      ;;; a label for the arc
     action_type       ;;; rotation or translation or combined
     action_param      ;;; rotation/screw axes or the direction
                       ;;; of translation
     action_range      ;;; the nominal range of rotation or
                       ;;; translation
     from_to_states    ;;; a pair of states the action starts
                       ;;; from and ends to
;
```

Each removed body in FAC has a record of its contact history. The collection of them all forms the POAMS.

The output of $\mathcal{KA}3$ POAMS contains more information than an ordinary task specification. POAMS has the following properties:

- the specification is unambiguous, complete and relevant to robot assembly actions;

- goal states are described in sufficient geometric and kinematic details;
- transitions between each pair of contact states are given;
- multiple choices for actions in the plan are provided.

5.4 Experiments on $\mathcal{KA}3$

$\mathcal{KA}3$ has been implemented in POPLOG on the SUN workstation under OS 4.0. Specificly, $\mathcal{KA}3$ is written mostly in POP11, although a few procedures are in POPLOG PROLOG. The system also has a POP11 interface to PADL2 which is in FORTRAN. Four assembly examples are presented in this chapter, each demonstrating a special aspect, to show how $\mathcal{KA}3$ solves them.

5.4.1 An Assembly with Double-pegs and Multiple Holes

We now examine how the planner finds mating features in the example of Figure 5.3 (page 119). This figure is drawn by PADL2 and extracted via snapshot. The surface labels are also provided by PADL2, but are typed on the figures by hand.

The input kinematic constraints are:

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

$\mathcal{KA}3$ starts its first stage: finding possible mating feature pairs. The process starts from the body that has the smallest number of surfaces. From bodies $b1, b2$ and $b3$, the same type of compound features $multi - insertor$ are found:

```
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]
```

The compound features that have the highest degree are kept to be matched first.

From body $b4$ compound features *multi-container* are found which have the same degree as what has been found so far.

```
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]
```

Now *KA3* starts to check the necessary mating feature conditions to prune out those pairs which are not eligible. Matching $b1$ and $b4$ 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 $b1$ *comp1* and $b4$ *comp1* is uniquely determined. The relative position of $b1$ to $b4$ is found to be:

```
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
```

With this position $\mathcal{KA}3$ can check the spatial constraint immediately, by calling the PADL2 *intersect* predicate

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

This says there is no spatial interference detected (Figure 5.4, page 120) Thus this node has satisfied the node consistency condition.

Similarly *comp*1 of $b1$ and *comp*13 of $b4$ are matched and spatial interference checking is also successful (Figure 5.5, page 121).

Matching $b2$ and $b4$ gives:

```
b2 comp1 and b4 comp1 are matched !
b2 comp1 and b4 comp13 are matched !
```

and matching $b3$ and $b4$ gives (Figure 5.6, page 122):

```
b3 comp1 and b4 comp12 are matched !
```

At this stage $\mathcal{KA}3$ has found all the compound features satisfying the spatial constraint locally. This means that a node consistent network has been achieved. The last step before each assembly configuration is confirmed to be feasible is to check the spatial constraint globally, which corresponds to checking an arc/path consistency. Figure 5.7 (page 123) shows that two interference volumes are found when the whole assembly is put together. Therefore the planner rejects this configuration. The process terminates since the domain of at least one of the kinematic constraints is empty.

After the user realizes the problem and provides a modified version of the in which the shapes of the bodies $b1, b2$ and $b3$ are changed example (Figure 5.8, page 124), the planner then confirms the following feasible assembly configurations in terms of compound mating features (Figure 5.9, page 125):

```
** [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 actually implies eight possible configurations of the given assembly since *comp*1 of $b1$ and *comp*1 of $b2$ both have the discrete rotational symmetries described at the end of Section 5.1.3. These reductions will be obvious when the output is re-written in terms of the primitive features composing each compound feature. Since $\mathcal{KA}3$ does not treat permutations of distinct bodies, this is the most compact characterization it can obtain.

The disassembly translational analysis of this example is straightforward, since the bodies do not interfere with each other. For each body, the direction of translation is uniquely determined. There are two contact states, not counting the one without any contacts, for bodies $b1$ (Similarly for $b2$ and $b3$):

```
s1==>
** <contact_state 1 gr_T1 [[fit [b1 3] [b4 10]]
                           [fit [b1 2] [b4 13]]] 1>
a1==>
** <preced_action 1 <procedure trans> <vec3 0 0 -1> 1 [s1 s2]>
```

```
s2==>
** <contact_state 2 gr_identity
                   [[fit [b1 3] [b4 10]]
                    [fit [b1 2] [b4 13]]
                    [fit [b1 1] [b4 7]]]
                   <false>>
```

The translation $a1$ given in the middle links the two states, a direction of motion is relative to the coordinate of the mating feature pairs. Notice that the changing of symmetry group from *gr_T*1 to *gr_identity* implies the changing of contact states for body $b1$, from the two-cylindrical-surface contact to its assembled position.

The total contact states an assembly, specified in terms of primitive features, together with the relative positions found during planning form the output of $\mathcal{KA}3$

5.4.2 An Assembly with All Revolute Joints

The assembly components of this assembly are shown in Figure 5.10 (page 126). The input assembly operations among bodies, implying three kinematic constraints, are:

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

The compound features of each body are composed of these primitive features:

```
b1 : comp1 = [7 8]
     comp2 = [9 10]
b2 : comp1 = [7 8]
     comp2 = [9 10]
b3 : comp1 = [7 9]
     comp2 = [7 8]
```

All the potential mating feature pairs are non-sharable. All the common symmetry groups of mating features are $SO(2)$, which corresponds to a revolute joint. Since $SO(2)$ is an infinite group, node/arc consistency checking in *step two* of stage two cannot be applied immediately. Although *comp*1 and *comp*2 of $b3$ are equivalent features, dismissing one would cause the process to terminate, so they are both preserved. During step three of stage two, the following assembly configurations are proposed after subgraph mapping:

```
** [Or  [And [fit [b3 comp1] [b1 comp2]]
             [fit [b3 comp2] [b2 comp2]]
             [fit [b1 comp1] [b2 comp1]]]
        [And [fit [b3 comp2] [b1 comp2]]
             [fit [b3 comp1] [b2 comp2]]
             [fit [b1 comp1] [b2 comp1]]]]
```

Although the exact relative position of $b3$ to $b1$ is unknown from their mating features, the symmetry constraint $SO(2)$ restricts $b1$ to having only one degree of freedom when it fits to $b3$: the rotation about the central axis of their mating features. As a matter of fact, every body in this assembly is constrained by two such revolute joints. Any pair of points coinciding on the mating feature surfaces should have the same orbit under the action of the common symmetry group of the feature pair. We picked the center point at the bottom of a cylinder, concave or convex, as the representative point for a cylindrical mating feature. This is a good choice for cylinders since its orbit under the symmetry group of the feature is itself (it is invariant under actions by any member of the symmetry group). Therefore the two representative points of a pair of mating features should be coincident all the time when the bodies fit (Figure 5.11, page 127). Figure 5.12 (page 128) shows that the orbit of the representative point on feature *comp*1 of $b2$ forms a locus under the symmetry group of feature *comp*2 of the same body, and the same is illustrated for the representative point of feature *comp*1 on body $b1$. The intersection of these two loci is the only possible position for the representative points of *comp*1 of $b1$ and *comp*1 of $b2$ to be coincident. Thus it uniquely determines the relative positions of bodies $b1$ and $b2$ such that both mating relationships can be satisfied simultaneously. Finally, the spatial constraint is applied and both sets of possible mating feature pairs are confirmed to be feasible assembly configurations.

The disassembly motion analysis is a bit more interesting than the previous example. Since only body $b3$ is movable, the contact states of $b1$ and $b2$ have to be *after* those of $b3$ in the disassembly process and thus *before* $b3$ while assembling. A temporal constraint is therefore established.

5.4.3 An Assembly with Both Discrete and Continuous Joints

There are two interesting points about the example shown in Figure 5.13 page 129:

One is that a pair of bodies in this assembly have a mating feature pair of triangular shape. Therefore the symmetry group of the feature is a cyclic group. For body $b3$, the symmetry group of its mating feature, the triangular shape compound feature, is the same as the symmetry group of the body itself. Thus the relative position of $b3$ and $b1$ is uniquely determined.

The other interesting point is that *KA*3 again has to use the method of loci to determine the relative positions of $b2$ to $b1$ and $b3$. This time it is the intersection of a point and a cylindrical surface.

The motion analysis is quite similar to the previous example.

5.4.4 A Gearbox

All the components in this gearbox are shown in Figure 5.14 page 130. There are four gears plus a base $b5$. The input assembly operation set is composed of:

```
the gear box with five parts
The Goal set ->
[goal 1 [fit b1 b5]]
[goal 2 [fit b2 b5]]
[goal 3 [fit b3 b5]]
[goal 4 [fit b4 b5]]
[goal 5 [mesh b1 b2]]
[goal 6 [mesh b1 b4]]
[goal 7 [mesh b2 b3]]
```

This example is one where not only *fit* but also *mesh* operations are incorporated into one plan. Note, mesh is not a surface contact mating, rather it is a line contacting one. At the stage of feature mating, when *KA*3 encounters the *mesh* operation, it checks first the "texture" of the surface and its corresponding parameters. For example,

```
 in mesh matching

surface textures:

tex_gear
tex_gear
[1 <cyl3 5.25 <ln3 0.0 -3.0 -8.0 0.0 0.0 1.0>>]
```

```
the symmetry group:

gr_cyclic

parameters checking:

5.25
17.25
13
43
 circular pitch are close enough

features 5  3 are meshed !!
```

The circular pitch is calculated using this formula:

$$\text{circular pitch} = \pi * (\text{pitch diameter } D)/\text{number of teeth } N$$

When a helical gear is present, the feature is associated with a symmetry group of a screw. Another example,

```
 in mesh matching

surface textures:
tex_helical_gear
tex_gear

no match, their symmetry groups are not compatible!
```

This matching fails because the symmetry groups are not compatible. Screw groups are considered as "black sheep" of the Euclidean group, and they only match with themselves.

Since in this example each body (except the base $b5$) has at least two contacts and one of them is mesh there is an extra checking for meshing after stage two. This is because only fitting relationships are guaranteed to be realized by the relative position analysis in Section 3.4. Before claiming it is a feasible assembly configuration, $\mathcal{KA}3$ makes sure that all the gear textured surfaces touch each other tangentially as required in the input kinematic constraints. This is the place where it is really convenient to have the boundary model of a solid around to identify the relevant surface and the bounding curves.

5.5 Discussion

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 [81].

Searching through the whole feature library can be inefficient when there are many entries. One way to organize the library is to use the degrees of compound feature 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 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 5.3 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 for the missing patterns to be recognized and added to the library after one trial. The current system does not have this learning ability. A desirable extension of this system is learning new, relevant, compound feature patterns to add to the library.

The work described here shows that an assembly planning system equipped with a feature library and the ability to represent and reason about symmetries of features and objects 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.

In this chapter we have also described the use of constraint satisfaction networks to reify assembly configurations which are composed of possible mating feature pairs related by symmetry constraints, i.e. the symmetry groups shared by the mating feature pairs. Through this work one can see that the performance of a special CSP depends strongly on how well the domain dependent subproblems have been solved. The domain dependent constraint application: finding possible mating features, finding positions, finding intersections etc., dominates the implementation as well. This seems to suggest that an effective combination of artificial intelligence and robotics must be woven together at each problem solving step.

The bottleneck of ACSP is the determination of the relative position of bodies so that the spatial constraint can be applied. The usefulness of symmetry constraints is reflected in the fact that when the identity group is present, the relative position can be computed uniquely. When no identity groups are present, we can apply the representative-point-intersection method to find possible locations or to claim that no solution exists. Things become more complicated when a symmetry constraint network is underconstrained, i.e. relative motions are allowed in an assembly for certain functional purposes, such as a pair of scissors. One plausible way to deal with this is simply to sweep the body concerned under the symmetry group of its constantly contacting features, then intersect this swept volume with the rest of the assembly. If no intersection is detected, the kinematic constraint is correctly instantiated, otherwise the intersected volumes set up boundaries (or intervals, label set [20]) on the kinematic constraint. PADL2 does not have the ability to sweep a body, therefore we have not yet experimented with this.

Our method cannot be complete even for assemblies with no residual degrees of freedom since it is possible to define assemblies that require the solution of a polynomial equation of arbitrarily high degree [36], and the geometric methods we use cannot generate solutions of such equations.

However in any practical assembly, only a small portion of the assembly planning problem will require the solution of such equations; the majority of the relationships should be amenable to analysis by our methods. Our system could be complemented by a system capable of treating a full range of polynomial equations as envisaged by Canny [13]. *KA3* has shown, however, 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.

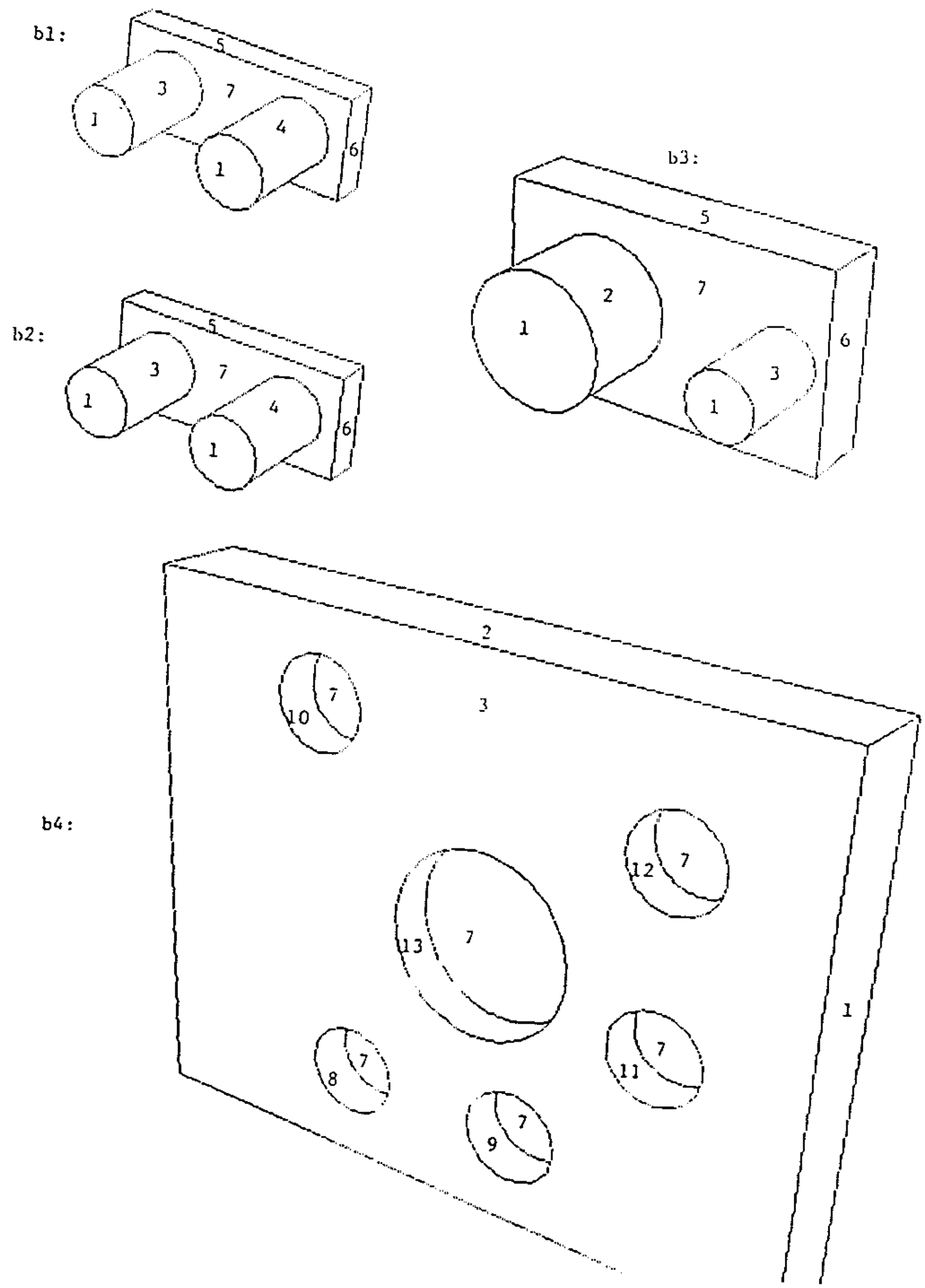

Figure 5.3: An assembly with double-pegs and multiple holes

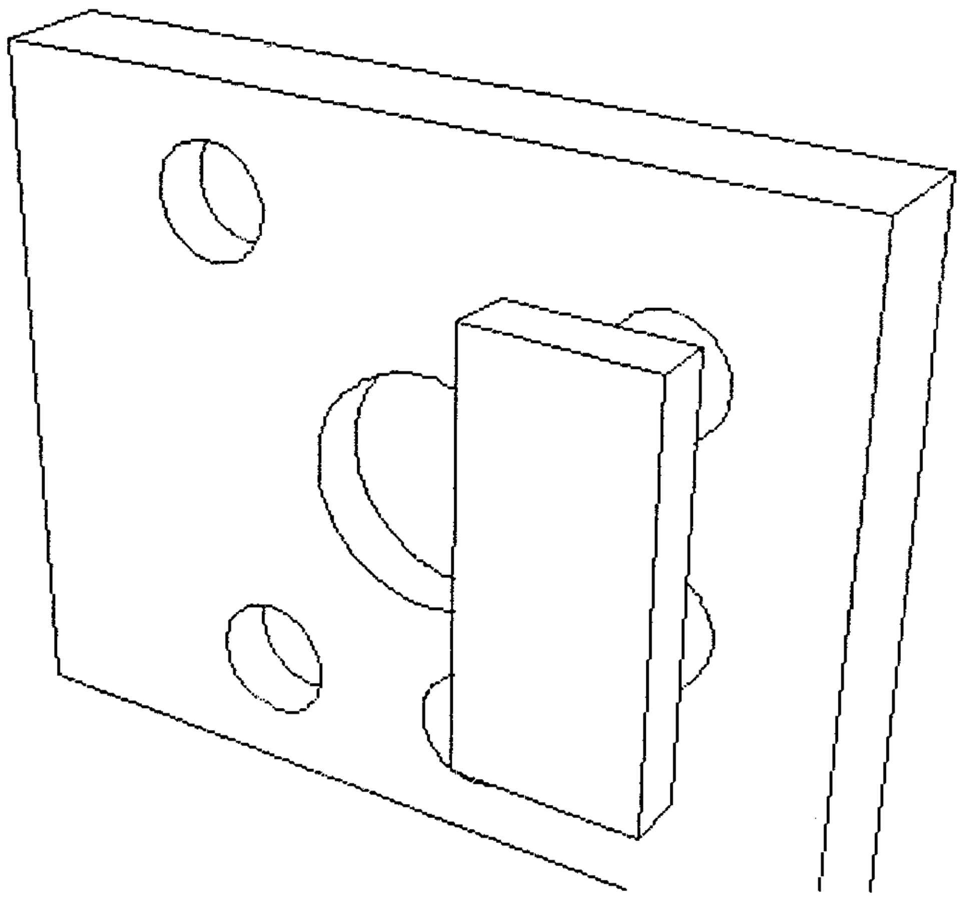

Figure 5.4: Body b1 and body b4 in one assembled position

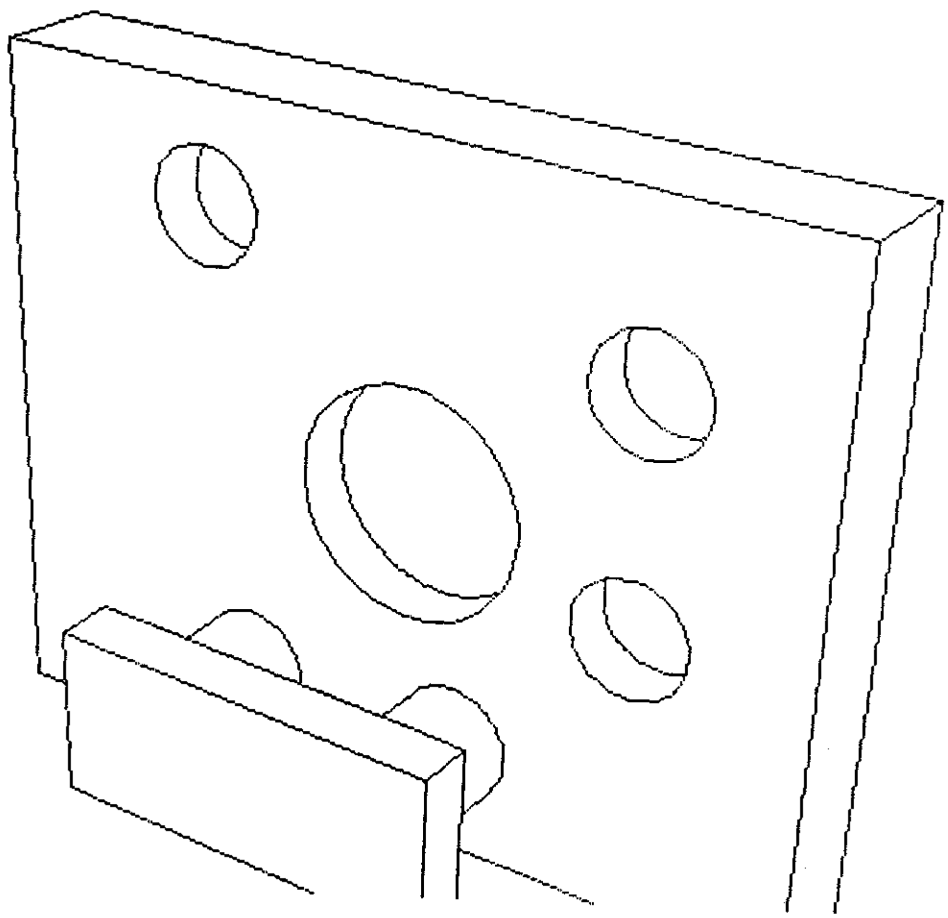

Figure 5.5: Body b1 and body b4 in another assembled position

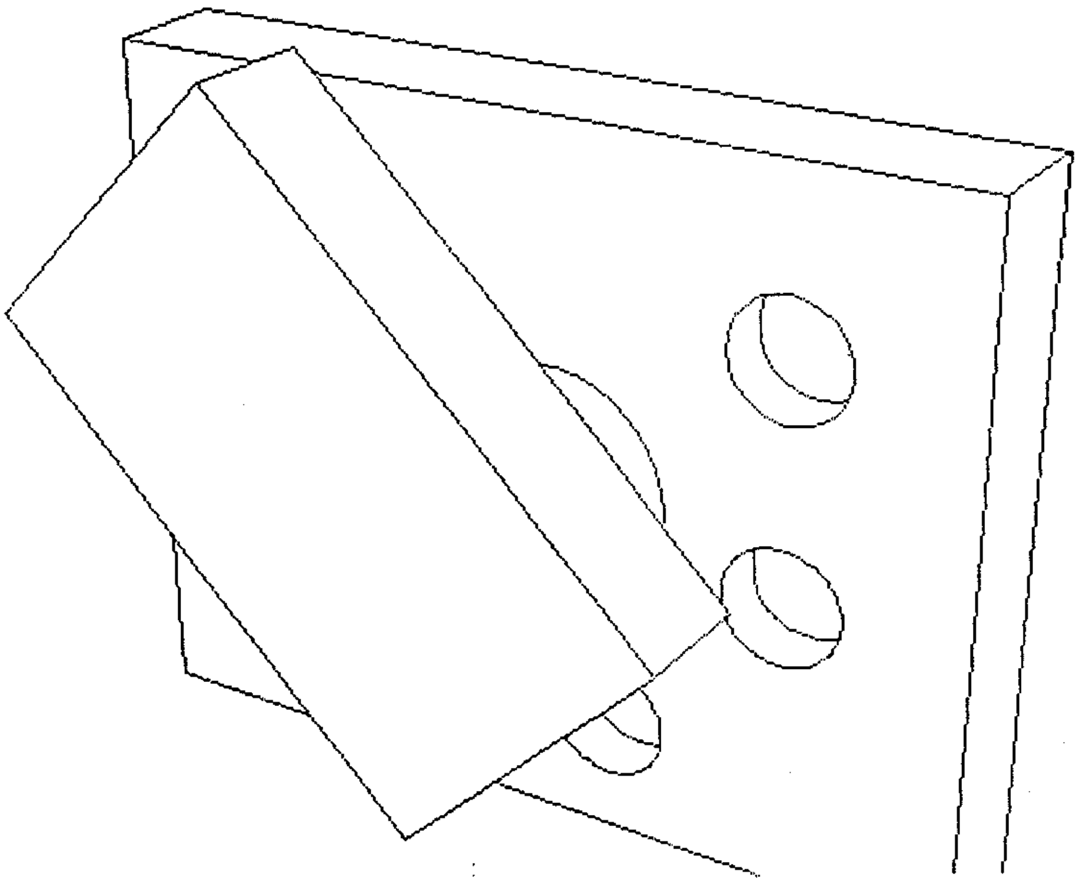

Figure 5.6: Body b3 and body b4 in an assembled position

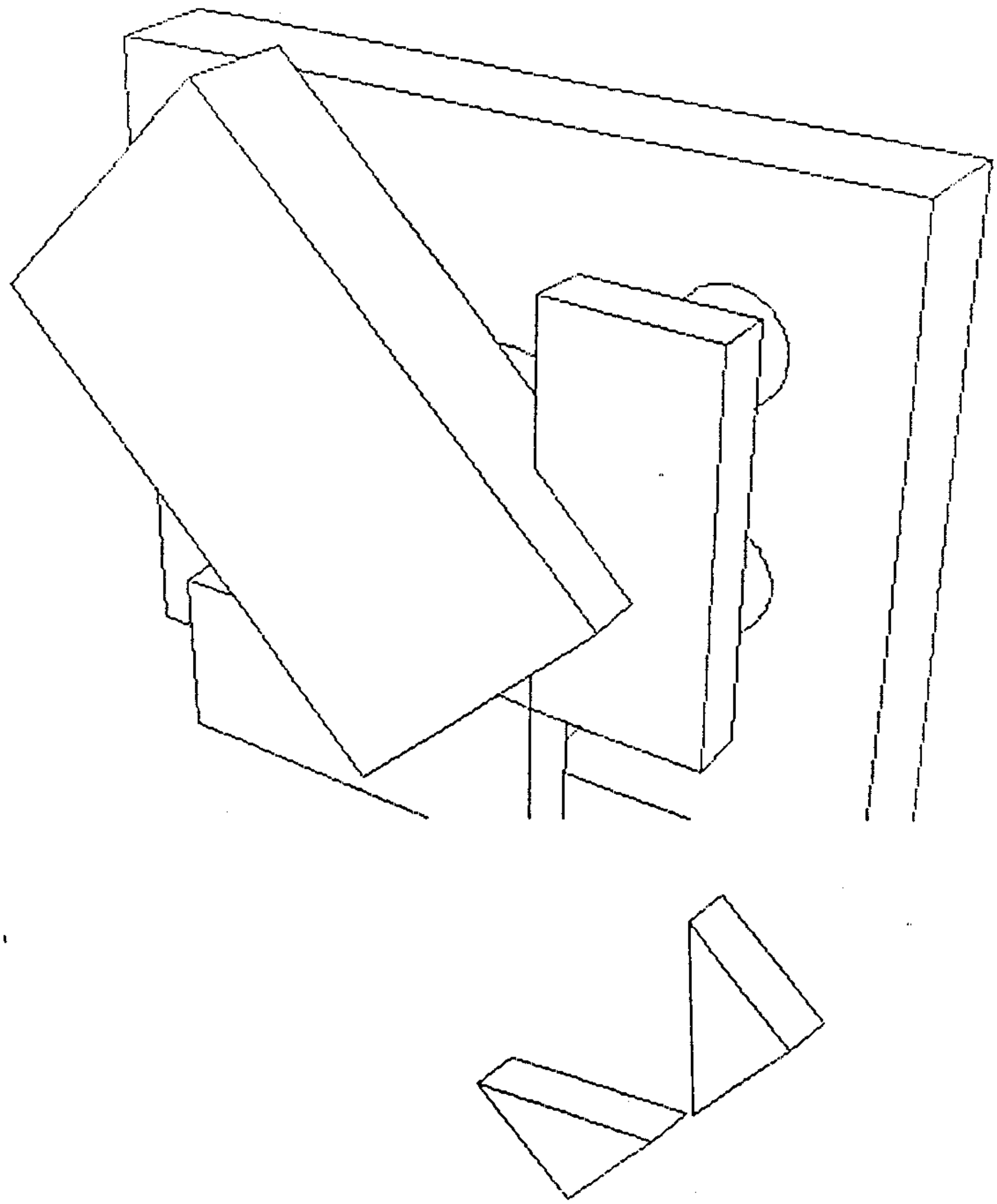

Figure 5.7: All four bodies in their assembled position — intersection detected

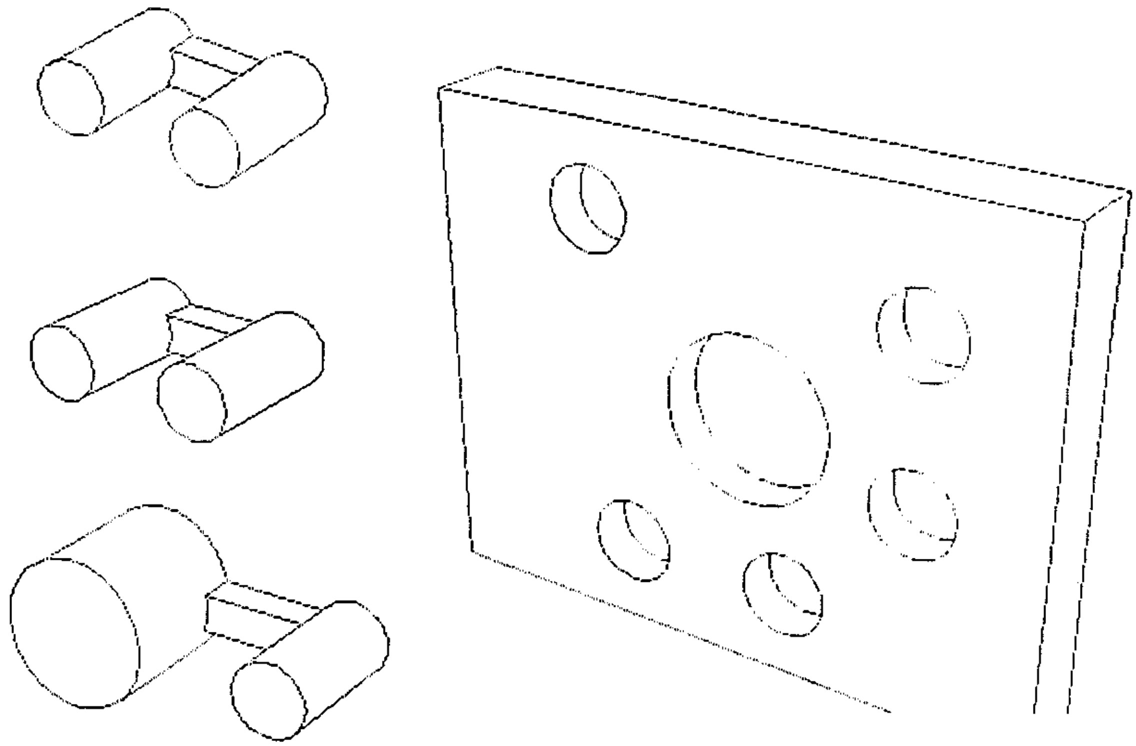

Figure 5.8: A modified version of the assembly example one

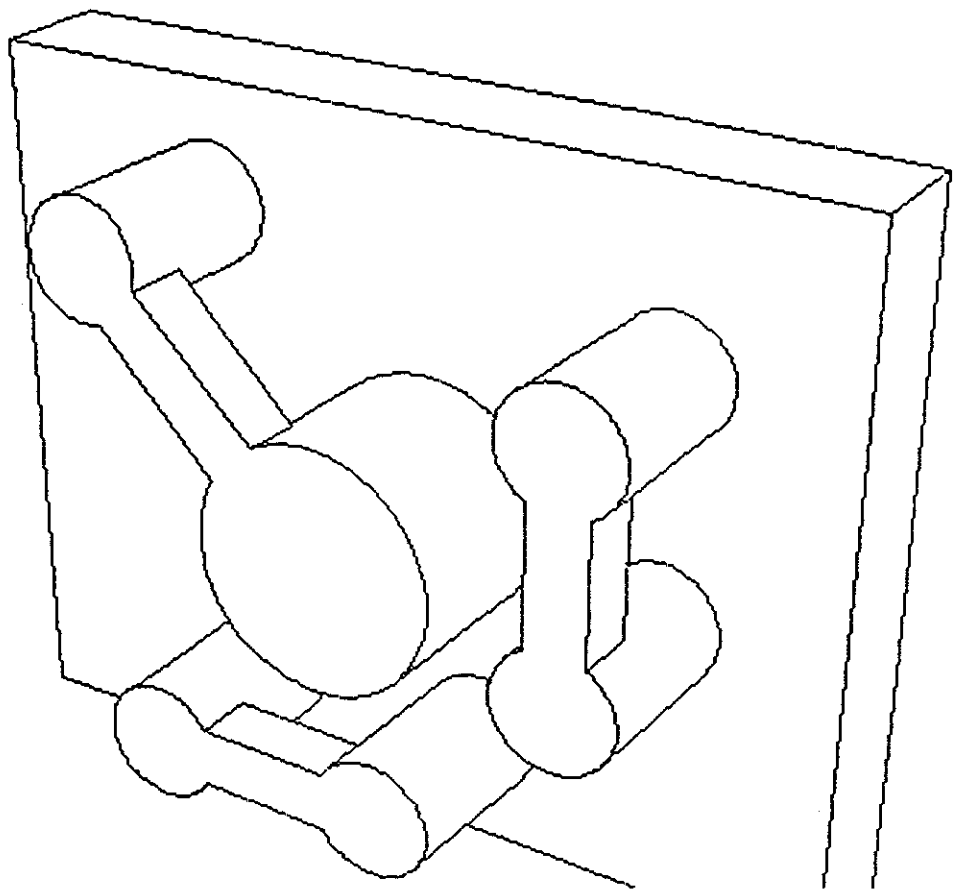

Figure 5.9: Revised four bodies in their assembled position

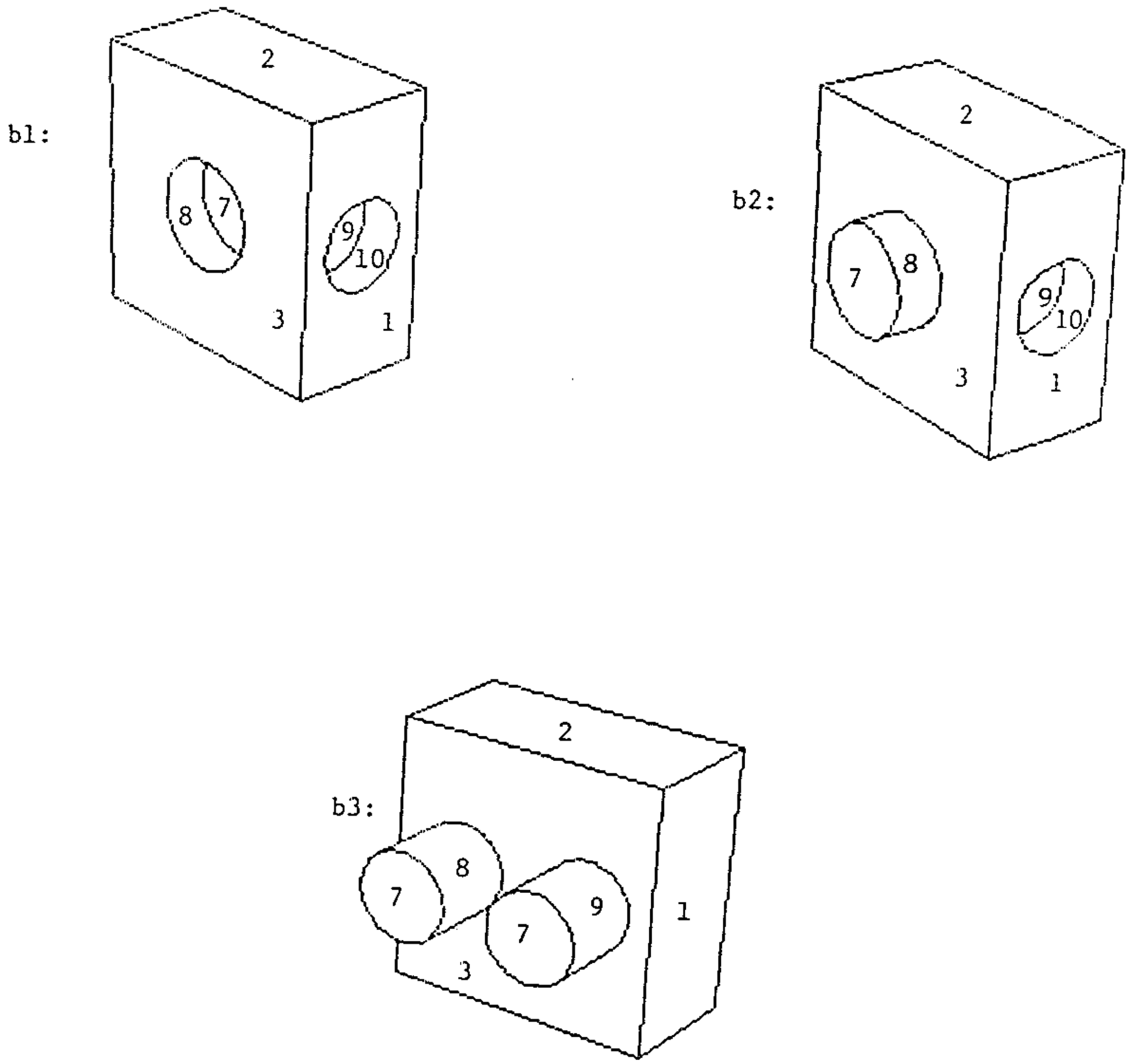

Figure 5.10: Second assembly example: an assembly with all revolute joints

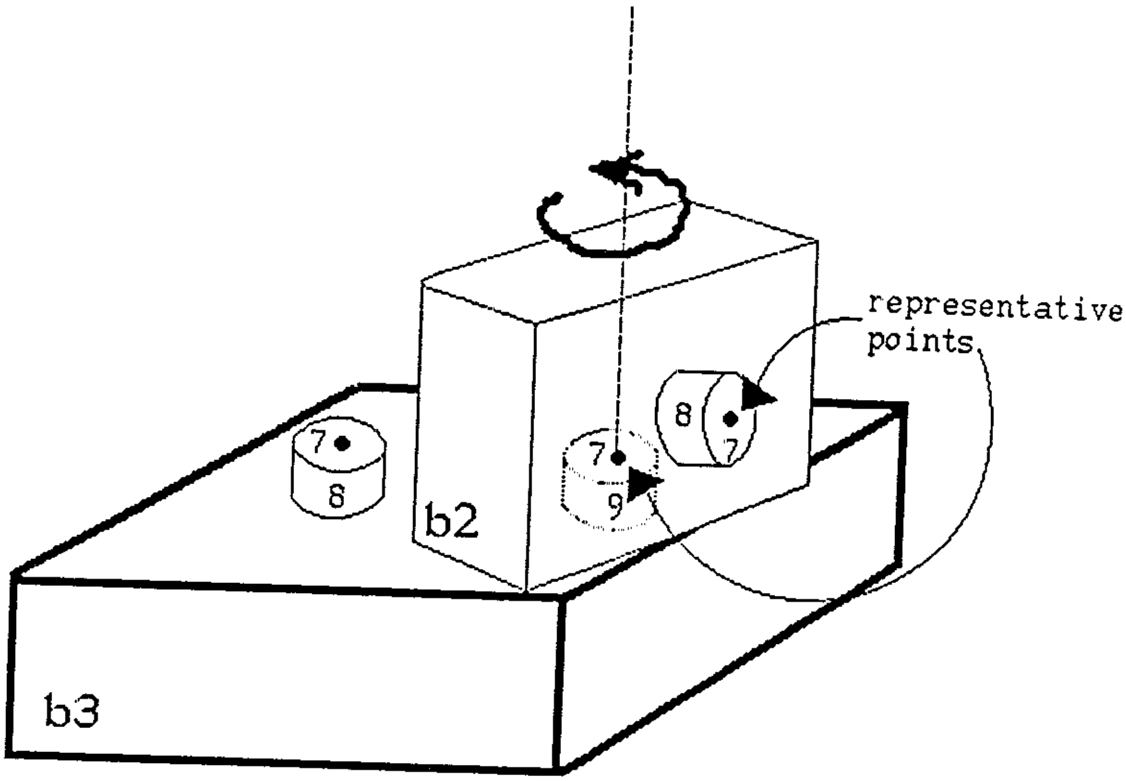

Figure 5.11: When $b2$ fits to $b3$, the representative points of the fitted features coincide, the orbit of the other representative point forms a locus under this revolute joint motion

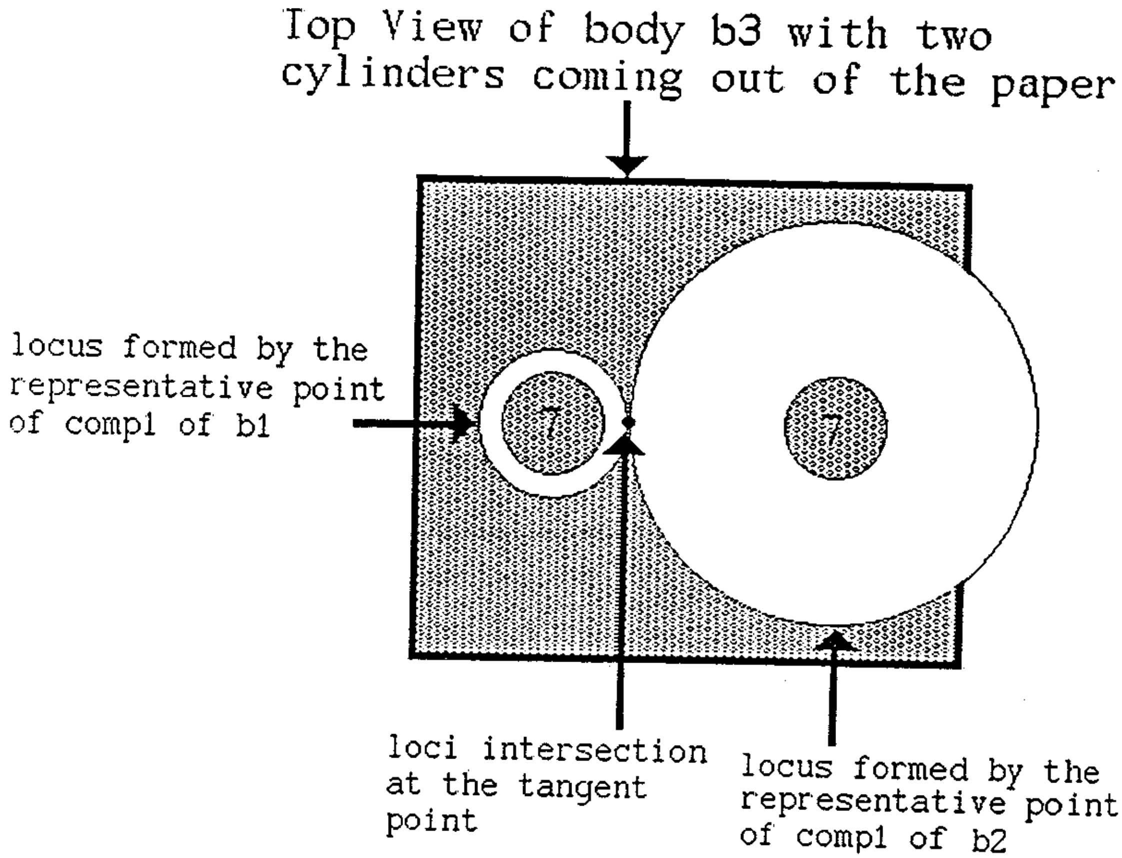

Figure 5.12: Intersecting loci from *compl* of *b1* and *compl* of *b2*

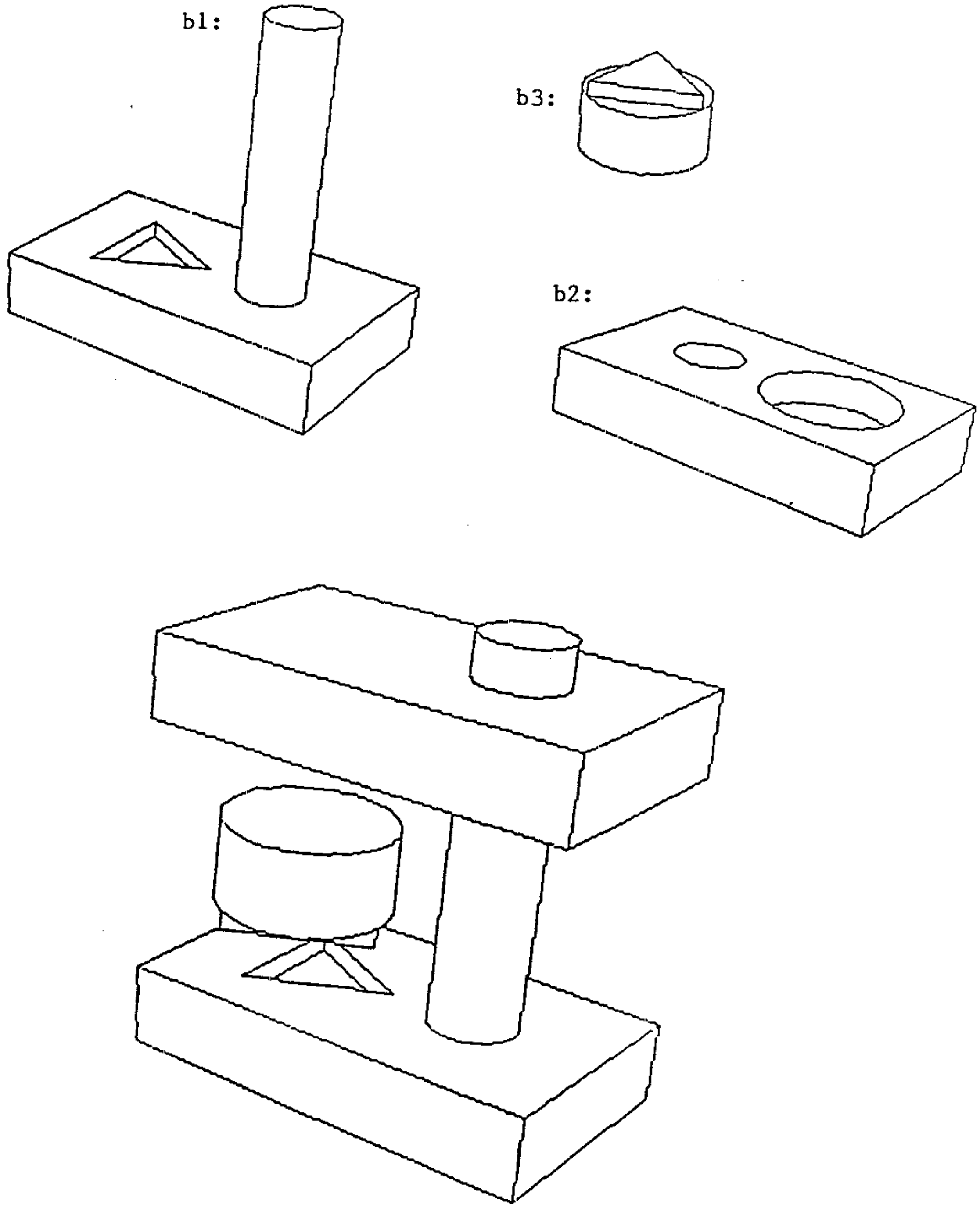

Figure 5.13: An example containing both discrete and continuous joints

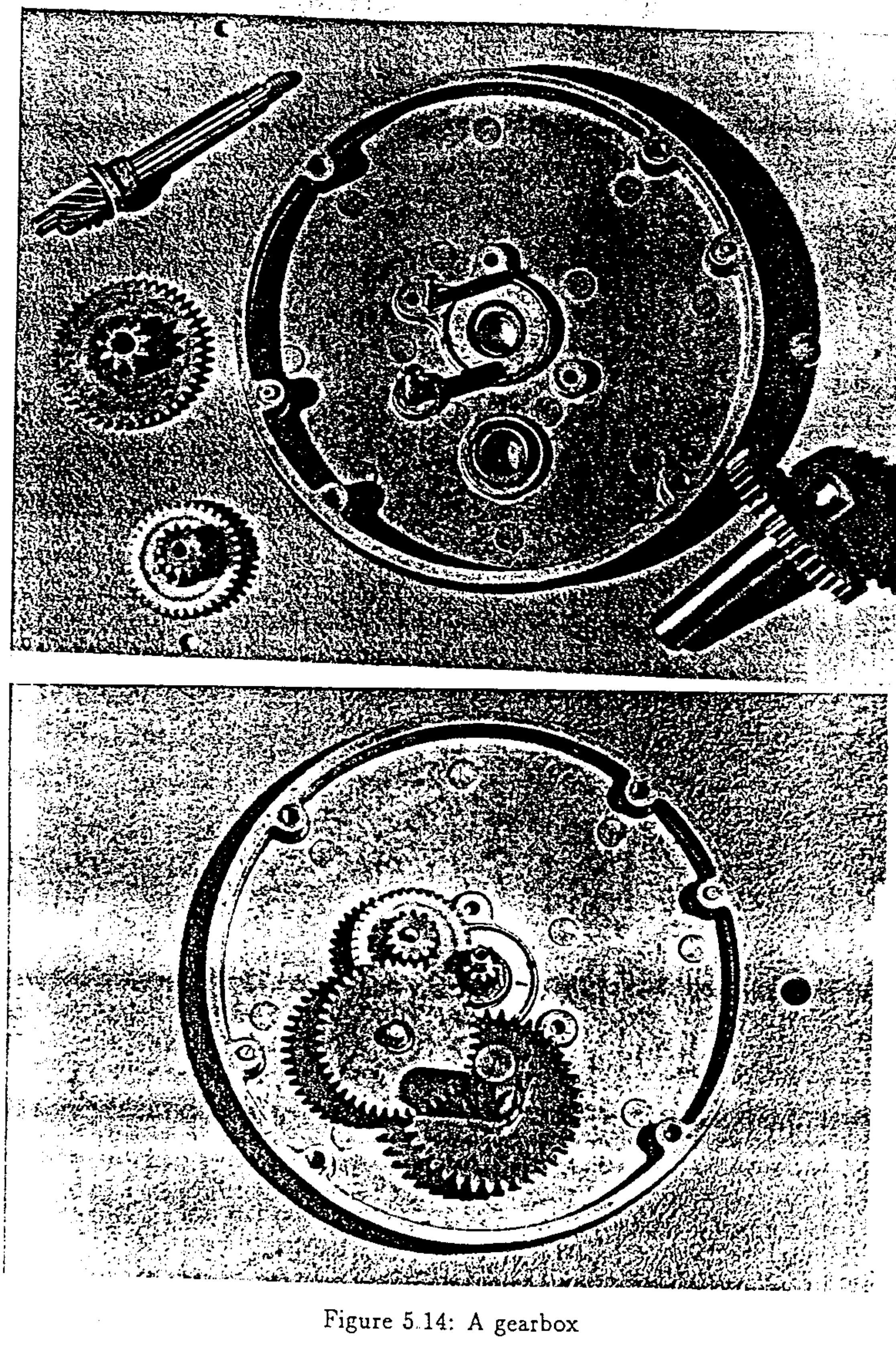

Figure 5.14: A gearbox

CHAPTER 6

CONCLUSION AND FUTURE WORK

This is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning.

— Winston Churchill

6.1 Contributions

The main contributions of this work are:

1) the formal establishment of the relations between a feature and its symmetry group, and between the symmetry group of a compound feature and the symmetry of its primitive features,

2) the geometric characterization and mathematical justification of TR subgroups of the proper Euclidean group, namely, the *characteristic invariants* representation,

3) the mathematical verification of the closure property of TR subgroups via intersection and the implementation of an efficient TR group intersection algorithm,

4) the development of a solid model-based kinematic assembly planning system $\mathcal{KA}3$ which combines rigorous treatment of symmetries of bodies and features, CSP problem solving techniques and heuristic search strategies into one computationally tractable framework, and

5) the avoidance of computationally expensive symbolic manipulation for finding solutions of algebraic equations, while still treating a set of cases which are common and important in assembly.

6.2 Future Research

This is all just a beginning. The limitations of and possible extensions to this work include:

- Only areal contact is formally treated. The product of **TR** groups is needed to extend this work to include both line and point contacts. Proposition 3.20 is a good place to start.
- $\mathcal{KA}3$'s ability is limited by the capability of the geometric modeller used.
- The geometric approach for finding relative positions by the intersection of loci method is incomplete; $\mathcal{KA}3$ could be complemented well by harnessing it to a system capable of treating a full range of polynomial equations.
- A fast algorithm is needed for finding body symmetries from boundary models that are not restricted to polyhedral, so that it will be possible to treat permutations of distinct bodies. Work reported in [85, 82, 30] is of relevance.
- The salient-feature-library is prestored and thus static; the ability to learn new features while planning is desired.
- $\mathcal{KA}3$ does not deal with subassemblies. To treat a robot hand grasping an object as assembly feature mating needs a proper characterization of subassemblies.

We treat robotic assembly planning as taking place at two distinct conceptual levels. Planning at the higher level involves deriving *nominal trajectories* along which the bodies to be assembled are to be moved. This is what $\mathcal{KA}3$ has done. Planning at the lower level transforms such a high-level specification into an assembly plan which takes account of uncertainty [19]. This is part of the on-going work in our lab.

Information in POAMS such as the d.o.f. between a pair of mating bodies, that is expressed in terms of symmetry groups of the mating features, can be translated into stiffness constraints for robot hand movements while assembling the mating bodies [66].

Sensory requests , as well as other kinds of requests (see below), can be embedded into the nominal plan POAMS as conditional statements. For example, an extra-action-request can be added into a plan to minimize position errors, especially at the critical contact states. A verification request can be added to keep the execution of the plan on the right track. The idea is to let the output plan contain as much contingency information as possible for on-line decision making.

The requests can be divided into several categories, such as:

- **minimal rotation request**: according to the symmetry properties of the mating bodies, give the maximum rotation angle needed to reach a mating state,
- **order request**: a sequence of actions,
- **verification request**: to verify some positive result(s) of actions,
- **error monitoring request**: request sensory feedback to ensure the robot is on the right track, and finally
- **alignment request**: actively minimize errors by adding extra actions.

Most of the above requests can be expressed in a formal syntax such as the RS model developed by Lyons in [54].

Stability analysis and selection of an optimal plan can also be carried out at this stage after both the kinematic and spatial constraints are satisfied. POAMS can also be translated directly into robot actions as displacement commands.

Information about assembly bodies can be gained from sources other than a geometric modeller; from range data, for example. $\mathcal{KA}3$ can then be used to find possible mating features to assemble an image of an object with its model. To do this, certain work needs to be done for dealing with noise in the data.

6.3 Remarks

In conclusion, the significance of this work is that it has addressed symmetries formally and computationally, especially the symmetries relevant to contact between bodies. Furthermore, it has embodied the theoretical results into a kinematic assembly planning system $\mathcal{KA}3$. $\mathcal{KA}3$ exhibits the interaction between algebra and geometry within a group theoretic framework and the interaction between CSP techniques and heuristic search strategies, providing us with a unified computational treatment of reasoning about how parts with multiple contacting features fit together.

This work has made it possible to realize

1) the automatic generation of assembly task specifications for robotic task level planning from the component diagrams of mechanical design, and

2) a robotic task level planner that 'understands' symmetry.

APPENDIX A

ABSTRACT ALGEBRA

Definition A.1 *Let $*$ be a binary operation defined on a set X and $\circ$ a binary operation defined on a set Y. A* **morphism** *or* **homomorphism** *is a function $f : X \to Y$ which 'carries' the operation from X to Y, i.e.*

$$f(a * b) = f(a) \circ f(b), \forall a, b \in X$$

A homomorphism $f : (X, *) \to (Y, \circ)$ is said to be

- *a* **monomorphism** *if f is injective;*
- *an* **epimorphism** *if f is surjective;*
- *an* **isomorphism** *if f is bijective.*

Definition A.2 *A* **linear transformation** *$f : V \to W$, of a vector space V to a vector space W over the same field F, is a transformation f of V into W which satisfies $f(x + y) = f(x) + f(y)$ and $f(c * x) = c * f(x)$ for all $x, y \in V$ and $c \in F$.*

Definition A.3 *A real quadratic form is said to be* **positive definite** *if, whenever x is no-zero, $q(x) = \sum_{i=1}^{n} k_i x_i^2 > 0$.*

Definition A.4 *The value $f(u, v) \in \Re$ is written $< u, v >$ and is known as the* **inner product** *of u and v. The inner product has the following properties for all $u, v \in V$ and $\lambda, \mu \in \Re$,*

- $< u, v > = < v, u >$ *(symmetry)*
- $< \lambda u_1 + \mu u_2, v > = \lambda < u_1, v > + \mu < u_2, v >$ *(bilinearity)*
- $u \neq 0 \Rightarrow < u, v > > 0$ *(positive definite).*

Definition A.5 *A vector space V over $\Re$ together with a symmetric bilinear form f whose associated quadratic form is positive definite is said to be an* **inner-product space**.

Definition A.6 *A linear transformation $g : U \to V$, where U and V are inner-product spaces, is* **orthogonal** *when*

$$< g(u), g(v) > = < u, v >, \forall u, v \in U$$

Definition A.7 *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.*

Theorem A.1 *If $m_1, \ldots m_k$ are positive integers that are pairwise relatively prime ($\gcd(m_i, m_j) = 1$ for $1 \leq i < j \leq k$) then for any integers $a_1, \ldots, a_k$ the system of congruences $y \equiv a_i \bmod m_i, i = 1, 2, \ldots, k$, has a simultaneous solution y that is uniquely determined modulo $m = m_1 \ldots m_k$. (Chinese Remainder Theorem)*

APPENDIX B

POINT-SET TOPOLOGY

These are some topology definitions and theorems taken from [18, 60]. They are listed here as a quick reference for the readers.

Definition B.1 *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 B.2 *The members of T are called* **open sets**.

Definition B.3 *A* **neighborhood** *of a point $x \in X$ is an open set containing x*

Definition B.4 *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 $\overline{X \backslash A}$.*

Definition B.5 *A function $f : X \to Y$ from a space X to a space Y is* **continuous** *provided that for each open set U in Y the inverse image*

$$f^{-1}(U) = \{x \in X | f(x) \in U\}$$

is open in X.

Definition B.6 *A one-to-one correspondence $f : X \to Y$ for which both f and the inverse function f^{-1} are continuous is called a* **homeomorphism**; *in this case X and Y are said to be* **homeomorphic**.

Definition B.7 *A function $g : X \to Y$ is* **open** *provided that $g(O)$ is open in Y for each open subset O of X.* **Closed function** *is defined analogously.*

Definition B.8 *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 B.9 *An n-***manifold** *is a separable metric space M^n in which every point has a neighborhood homeomorphic to $\Re^3$.*

Definition B.10 *Two sets H, K are* **separated** *if*

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

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

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

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

APPENDIX C

FINITE ROTATION GROUPS

If the rotation group G has order n, then $n = m_i n_i$ where m_i is the number of *poles* in the *orbit* Γ_i and n_i denotes the order of Z_i, which is the *stabilizer* subgroup of any *pole* in G. The number of rotation axes through the center O of a solid is $(\Sigma\ m_i)\ /2$. Table-C.1 is a complete list of finite rotation symmetry subgroups of $SO(3)$. For any bounded solid, the rigid motions which bring it to its original 3-d space form a symmetry group and the group will be one of the finite symmetry groups listed in Table-C.1. For example, associated with $block(1,2,3)$ is a dihedral group with 3 rotational axes and each element of the group has order 2; and associated with $cyl(2,6)$ is an $O(2)$ group.

Table C.1: Types of finite rotation groups

Group Name	$\lvert G \rvert$ size	Poles in Orbit 1	Poles in Orbit 2	Poles in Orbit 3	Comments
identity	1	∞	0	0	the only rotation leaves than two points of Σ fixed
cyclic	n	1	1	0	generated by the $2\pi/n$ rotation
dihedral	$n = 2m$	$n/2$	$n/2$	2	for a regular m-gon, $m+1$ rotation axes
tetrahedral	12	6	4	4	7 rotation axes
octahedral	24	12	8	6	13 axes, same as cube
icosahedral	60	30	20	12	26 axes, same as dodecahedron

APPENDIX D

INTERSECTION OF DISCRETE TR GROUPS

Let $G_1 = T_1R_1, G_2 = T_2R_2$. When both T_1, T_2 are discrete there are two possible cases for $G_1 \cap G_2$ to have a non-trivial intersection:

- the two lattices are overlapping (parallel)
- the two lattices are perpendicular to each other

Algorithm:

Firstly, find $T_1 \cap T_2 = \beta_t^{-1}(\mathcal{T}_{T_1}, \mathcal{T}_{T_2})$. If $T_1 \cap T_2 = \{1\}$ then $G_1 \cap G_2$ is simply a rotation subgroup or the identity group.

For two one-dimensional lattices:

By definition of the translational invariants, at least $s_0 \in \mathcal{T}_{T_1} \cap \mathcal{T}_{T_2}$.

Suppose generators for T_1, T_2 are t_1, t_2 respectively. If $t_1 \| t_2$ and $lcm(||t_1||, ||t_2||) = t$ exists then t, which is parallel with both t_1 and t_2, is the generator of $T_1 \cap T_2$. If the two lattices $\mathcal{T}_{T_1}, \mathcal{T}_{T_2}$ do not coincide anywhere else except the origin s_0 then the intersected translation group is the identity group.

Two two-dimensional lattices:

Let a, b be generators of T_1, and c one of the generators T_2. Since it will be clearly said, a generator, a translation, and its norm will not be labeled differently in the following statement. Suppose θ is the angle between c and a (Figure D.1). Then

$$tan(\theta) = \frac{xb}{ya}$$

If $\frac{a}{b}tan(\theta)$ is a rational number then it can be rewritten as a reduced fraction $\frac{p}{q}$ where p, q are integers. Therefore $x = p, y = q$ is the solution. This only indicates that lattice of T_1 has points residing on the line with $tan(\theta)$ with interval $d = \sqrt{(xb)^2 + (ya)^2}$. Now to find out whether lattices of T_1 and T_2 have any intersections is to find out whether the one dimension lattice with interval d and the one-dimension lattice generated by c of T_2 have a lcm as we did with two one-dimensional lattices above. Similar approach can be applied to the other generator of T_2. The result of this step is to find $T_1 \cap T_2$ which can be zero, one or two dimensions. This gives the dimension(s) of a 'window' used in the following step to find rotation subgroup of the intersection.

Secondly, find all the rotation subgroups within such a *window*: the unit lattice of the intersected group. This window can be placed by any translation but obviously there is an optimal position to locate the corner of the window i.e. the highest-order rotation subgroup of the intersected group. In this case only the rotations at the corners of the window need to be checked out.

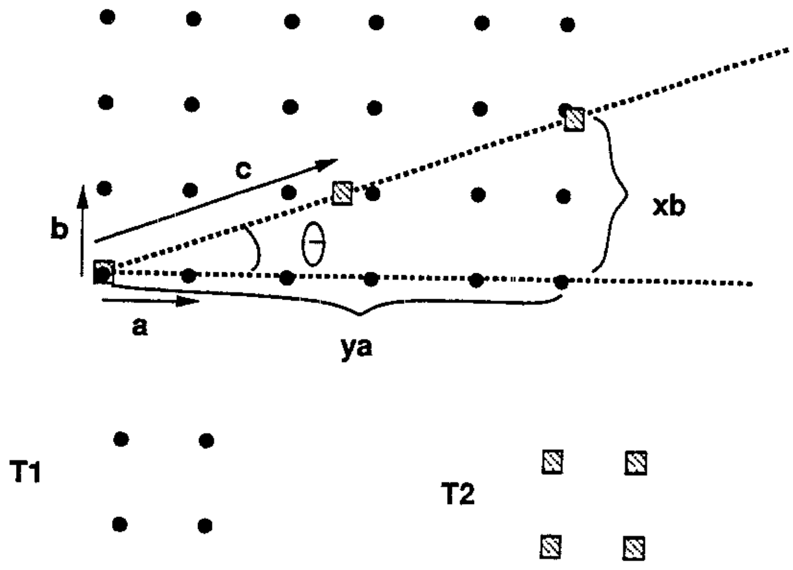

Figure D.1: Two overlapping lattices

When we calculate the rotations, we encounter the case where two lattices (fixed point sets) do not have any point in common at s_0. A more general method using Chinese Remainder Theorem is the following:

Suppose each lattice has an off-set, one is a_0 the other is b_0, the norm of generator is a, b respectively. Given integers x, y, question: is there a solution for the following equation?

$$a_0 + ax = b_0 + by \Rightarrow ax - by = b_0 - a_0 = c$$

The condition for this equation to have a solution is:

$$\gcd(a, b) | c$$

As far as the original offsets and norms are rational numbers, this Diophantine equation will have a solution.

Thirdly, find the rotation subgroup R with the highest order in the window. There are only finite number of them to calculate: the ones at the corner of the window.

Finally, $(T_1 \cap T_2)R = G_1 \cap G_2$ is the intersected group.

This algorithm has not been implemented, mainly because of the difficulty of telling whether an expression, such as $\tan(\theta)$, is a rational number precisely.

APPENDIX E

IMPLEMENTATION OF SHAPES USING PADL2

A geometric solid modeller is a software package providing informationally complete representations of solids such that any well-defined geometrical property of any represented solid can be calculated automatically [70, 71]. PADL2 [11] is such a solid modeller, and we have made use of it to provide us with a *boundary representation* of solids in terms of faces, edges and vertices. Solid shapes, specified as Prolog terms in the formalism used in the Edinburgh Designer System [64], are converted to the PADL2 input formalism, and processed by that system into boundary models.

In the volumetric or Constructive Solid Geometry (CSG) approach to defining shapes, nominal shapes are defined as subsets of $\Re^3$ by taking a collection of *primitive shapes*, relocating them with rigid transformations, and combining them with the boolean operations of union, intersection and set-difference.

The bounded shapes that can be described symbolically for PADL2 have the following form as Prolog terms.

```
block(X,Y,Z)     -- a block with dimensions X,Y,Z
cyl(H,R)         -- a cylinder of height H and radius R
sph(R)           -- a sphere of radius R
wed(X,Y,Z)       -- a wedge with dimensions X,Y,Z
con(H,R)         -- a cone of height H and radius R
tor(Min,Maj)     -- a torus with minor and major radii Min and Maj
```

Primitive or composite shapes may be combined through the following boolean operations:

```
Shape1 \/ Shape2  -- union of Shape1 and Shape2
Shape1 /\ Shape2  -- intersection of Shape1 and Shape2
Shape1 \ Shape2   -- difference of Shape1 and Shape2
```

Rigid transformations are specified using the following Prolog rendering of the forms.

```
trans(X,Y,Z)        -- a pure translation along the x,y, and z-axes
trans(vec(X,Y,Z))                  (same)
rot(ii,T)           -- a rotation of T radians about the x-axis
rot(jj,T)           -- a rotation of T radians about the y-axis
rot(kk,T)           -- a rotation of T radians about the z-axis
rot(vec(X,Y,Z),T)   -- a rotation of T radians about a directional
                       vector %(which need not be normal)
```

Table E.1: Infinite Solids and their descriptions

Solid	Defining Constraint	Description
H	$\{(x,y,z) \mid z \leq 0\}$	Infinite half-space
Cyl(r)	$\{(x,y,z) \mid x^2+y^2 \leq r^2\}$	Infinite cylinder, radius r
Sph(r)	$\{(x,y,z) \mid x^2+y^2+z^2 \leq r^2\}$	Sphere, radius r
Cone(α)	$\{(x,y,z) \mid x^2+y^2 \leq z \tan\alpha, z \geq 0\}$	A cone, angle α
Screw(p, f)	$x^2+y^2 \leq f(\frac{z+p\arg(x+iy)}{2\pi} mod\ p)$	Screw, pitch p and profile f
Gear(n, r, f)	$x^2+y^2 \leq r^2 + f(\arg(x+iy) mod\ \frac{2\pi}{n})$	Gear, n-teeth, radius r profile f

If *Shape* is a shape, and *Loc* is a rigid transformation, then *Shape @ Loc* is *Shape* relocated by *Loc*.

For example

```
cyl(4,1)  \/ (cyl(4,1)@trans(4,0,0)) \/
   (block(1,2,6)@trans(-1,-1,1)@rot(jj,1.570796)) )
```

denotes an object, consisting of two cylinders "stuck" on to a block with the union operation \/.

A *draw* predicate is provided which "prints" the Prolog term as a string in PADL2 syntax, and then calls PADL2 to form and display a boundary model of the object. The PADL2 internal representation of the boundary model is extracted using FORTRAN subroutines linked into POPLOG [29, 78] as external procedures; POP-11 objects expressing the face-edge-vertex structure of a body are built from the information thus extracted. These objects correspond quite closely with the structures that PADL2 uses internally.

Table E.1 gives the definition of some infinite solids. The boundaries of these solids correspond with the surfaces used by PADL2, except for the *Screw* and *Gear* forms. We treat these in our system by 'lying' to PADL2 — they exist in our input formalism, they are approximated by cylinders before input to PADL2, and relabeled in the resultant boundary representation.

APPENDIX F

DEFINITION FOR EDGES AND GENERAL FEATURE TYPES

Definition F.1 Edge: *an* edge *of a solid is an intersection of two distinct adjacent surfaces of a solid.*

Definition F.2 *The* **inherent direction** *of a edge itself is pre-defined arbitrarily However when an edge bounds a surface P the* **direction** *of the edge at point p is $\vec{e}_p$ and defined as :*

$\vec{e}_p \times \vec{l}_p = \vec{n}_p$ *where $\vec{l}_p$ is the vector pointing from point p into the surface P and $\vec{n}_p$ is the surface normal at point p.*

The direction of the edge is defined as such that when traversed along it the surface is always on the left hand of the edge.

Definition F.3 Convex and concave points on an edge: *if an edge e has an inherent direction $\vec{e}_p$, the point p on the edge is said to be a* convex *point if and only if*

$$(\vec{l} \times \vec{r}) \cdot \vec{e_p} > 0$$

where when the body is observed from outside the material and with $\vec{e_p}$ pointing upwards, $\vec{l}$ and $\vec{r}$ are the normal vectors of the surfaces on the left and right of $\vec{e_p}$ respectively.

A concave *point is defined correspondingly*

Definition F.4 Convex and concave edges: *A* convex *edge is an edge which is composed of* convex points *only. A* concave *edges is defined similarly.*

Note some edge can be partially convex and partially concave (two cylindrical surfaces with their central axes intersect perpendicularly).

Definition F.5 Container and insertor: *a compound feature is said to be a* container *if every adjacent primitive feature pair shares a* concave edge; *it is said to be an* insertor *if every adjacent primitive feature pair shares a* convex edge.

APPENDIX G

WHY IS THE PLANNING SYSTEM CALLED *KA3*?

This name comes from a Chinese word: *KA3* where *KA* denotes the sound and 3 denotes the third tone out of four choices. I have long been thinking of a name for this system but none was suitable. One day, I was thinking of the role this system could play in the design-to-manufacturing spectrum, "It acts like an interface which connects the designers, at a high level in this spectrum, and the robotics task-level planner, which is relatively lower in this spectrum", naturally, a Chinese phrase was jotted on the paper: *cheng2 shang4 qi3 xia4* (Figure G.1). It means something or someone that takes the consequence from the above and invokes the below in an active and complementary manner. In Chinese "upper" or "above" is pronounced as: *shang4* and "lower" or "below" is: *xia4*. The interesting thing is that when I let the two words *shang4* (upper) and *xia4* (lower) "interface" with each other it forms another Chinese word: *KA3* (Figure G.1). One of the main meanings of *KA3* is *interlock*, which is what assembly is all about.

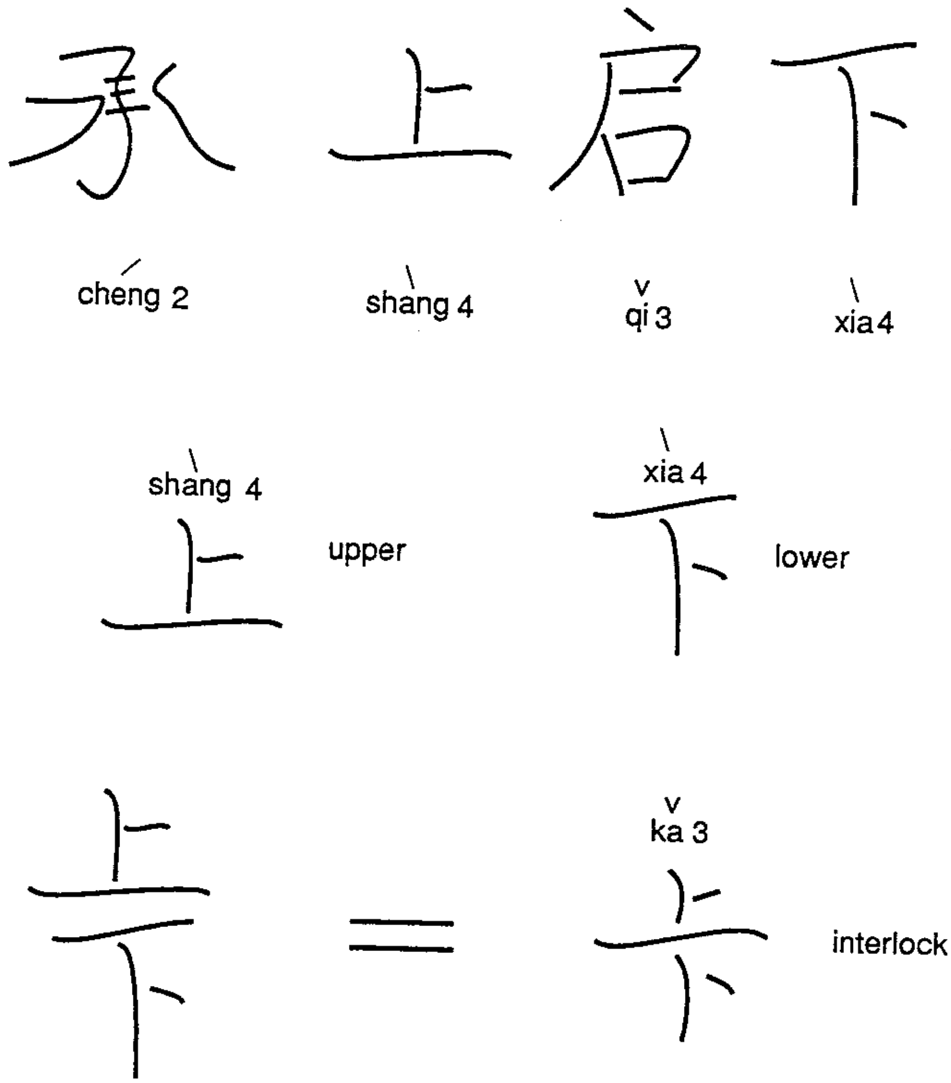

Figure G 1: A Chinese phrase: *cheng shang qi xia* meaning: interfacing the upper and the lower. The word *upper* (shang 4), *lower* (xia 4) and *interlock* (ka 3) in Chinese

BIBLIOGRAPHY

[1] A.P. Ambler and R.J. Popplestone. Inferring the positions of bodies from specified spatial relationships. *Artificial Intelligence*, 6:157–174, 1975.

[2] J. Angeles. *Spatial Kinematic Chains: Analysis - Synthesis - Optimization*. Springer-Verlag, New York, 1982.

[3] J.G.F. Belinfante and B. Kolman. A survey of Lie groups and Lie algebra: with applications and computational methods. In *SIAM*, Philadelphia, 1972.

[4] G. Birkhoff and S. Mac Lane. *A Survey of Modern Algebra*. Macmillan Publishing Co., Inc., New York, fourth edition, 1977.

[5] G. Boothroyd and P. Dewhurst. Design for assembly - a designer's handbook. Technical report, Department of Mechanical Engineering, University of Massachusetts, Amherst, MA., 1983.

[6] G. Boothroyd, C. Poli, and L.E. Murch. Handbook of feeding and orienting techniques for small parts. University of Massachusetts, Amherst, MA.

[7] G. Boothroyd, C. Poli, and L.E. Murch. *Automatic Assembly*. Marcel Dekker, Inc., New York, 1982.

[8] A. Bourtjault. *Contribution a une Approche Méthodologique de L'Assemblage Automatisé: Elaboration Automatique des Séquences Opératoires*. PhD thesis, Université de Franche-Comté, Besançon, France, November 1984.

[9] M. Brady. Artificial intelligence and robotics. *Artificial Intelligence*, 26:79–121, 1985.

[10] A.R. Brooks. Symbolic reasoning among 3-d models and 2-d images. *Artificial Intelligence*, 17:285–348, 1981.

[11] C.M. Brown. PADL2: A technical summary. *IEEE Computer Graphics and Applications*, 2(2):69–84, March 1982.

[12] S. Cameron. *Modelling Solids in Motion*. PhD thesis, University of Edinburgh, Edinburgh, U.K., 1984.

[13] J. Canny. *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, MA., 1988.

[14] P.C. Carney and Brown D.C. A continued investigation into qualitative reasoning about shape and fit. submitted to AI EDAM, November 1988.

[15] D. Chapman. Planning for conjunctive goals. *Artificial Intelligence*, 32:333–377, 1987.

[16] P.R. Cohen and E.A. Feigenbaum, editors. *The Handbook of Artificial Intelligence*, volume 3, chapter XV, pages 513–556. William Kaufmann, Inc., Los Angelos, CA., 1982.

[17] H.S.M. Coxeter and W.O.J. Moser. *Generators and Relations for Discrete Groups*. Springer-Verlag, New York, fourth edition, 1980.

[18] F.H. Croom. *Basic Concepts of Algebraic Topology*. Springer-Verlag, New York, 1978.

[19] G. Dakin and R.J Popplestone. Computing applied force constraints for insertion tasks using halfspace intersection projections. Technical Report COINS TR90-56, Department of Computer and Information Science, University of Massachusetts, Amherst, MA., 1990.

[20] E. Davis. Constraint propagation with interval labels. *Artificial Intelligence*, 32(3):281–331, 1987.

[21] R.L. De Fazio and D.E. Whitney. Simplified generation of all mechanical assembly sequences. *IEEE Journal of Robotics and Automation*, RA-3(6), December 1987.

[22] R. Dechter and J. Pearl. Network-based heuristics for constraint satisfaction problems. *Artificial Intelligence*, 34(1):1–38, 1987.

[23] J.R. Dixon and C.L. Dym. Artificial intelligence and geometric reasoning in manufacturing technology. *Appl. Mech. Rev*, 39(10), 1986.

[24] B.A. Dubrovin, A.T. Fomenko, and S.P. Novikov. *Modern Geometry - Methods and Applications, part I. The Geometry of Surfaces, Transformation Groups, and Fields*. Springer-Verlag, New York, 1984. Gehring, F.W., Halmos, P.R. and Moore, C.C. (eds).

[25] R.E. Fikes and N.J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 3:251–208, 1971.

[26] K.S. Fu, R.C. Gonzalez, and C.S.G. Lee. *ROBOTICS: Control, Sensing, Vision, and Intelligence*. McGraw-Hill Book Company, 1987.

[27] J. A. Gallian. *Contemporary Abstract Algebra*. D.C. Heath and Company, 1986.

[28] D.S. Green and D.C. Brown. Qualitative reasoning during design about shape and fit: a preliminary report. In J.S. Gero, editor, *Expert Systems in Computer-Aided Design*, pages 93–112. North-Holland, Amsterdam, 1987.

[29] S. Hardy. A new software environment for list-processing and logic programming. In T. O'Shea and M. Eisenstadt, editors, *Artificial Intelligence, Tools, Techniques and Applications*. Harper and Row, New York, 1984.

[30] A. Helmut, K. Mehlhorn, H. Wagener, and E. Welzl. Congruence, similarity, and symmetries of geometric objects. *Discrete and Computational Geometry*, 3(3):237–256, 1988.

[31] M.R. Henderson. *Extraction of Feature Information from Three Dimensional CAD Data*. PhD thesis, Purdue University, 1984.

[32] N.F.M. Henry and K. Lonsdale, editors. *International Tables for X-ray Crystallography, Volume 1, Symmetry Groups*. The Kynoch Press, England, 1969. The International Union of Crystallography.

[33] J.M. Hervé. Analyse structurelle des mécanismes par groupe des déplacements. *Mechanism and Machine Theory*, 13(4), 1978.

[34] L.S. Homem de Mello. *Task Sequence Planning for Robotic Assembly*. PhD thesis, Carnegie Mellon University, 1989.

[35] L.S. Homem de Mello and A.C. Sanderson. AND/OR graph representation of assembly plans. In *The Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 1113–1119. American Association for Artificial Intelligence, Morgan Kaufmann Publishers, 1986.

[36] J.E. Hopcroft, D. Joseph, and S. Whitesides. Movement problems for 2-dimensional linkages. In J.T. Schwartz, M. Sharir, and J. Hopcroft, editors, *Planning, Geometry and Complexity of Robot Motions*. Ablex Pub. Corp, Norwood, N.J., 1987.

[37] J.E. Hopcroft and G.T. Wilfong. On the motion of objects in contact. In *Proceedings of the 2nd International Symposium on Robotics Research*, pages 81–90, Kyoto, Japan, 1984.

[38] S.A. Hutchinson and A.C. Kak. SPAR: A planner that satisfies operational and geometric goals in uncertain environments. *Artificial Intelligence Magazine*, Spring 1990.

[39] D.E. Jakopac. *A Multi-level, Model Based Planner for Mechanical Assembly*. PhD thesis, Northwestern University, Evanston, Illinois, June 1985.

[40] F. Klein. *Lectures on the Icosahedron*. Dover, New York, 1957. English Translation, Chap. 5, Sec. 2.

[41] A. Koutsou. *Planning Motion in Contact to Achieve Parts Mating*. PhD thesis, University of Edinburgh, Edinburgh, U.K., 1986.

[42] G.A. Kramer. Solving geometric constraint systems. In *Proceedings of AAAI-90*, Boston, MA., 1990.

[43] C. Laugier. Planning fine motion strategies by reasoning in the contact space. In *IEEE International Conference on Robotics and Automation*, pages 653–661, Washington, D.C., 1989. IEEE Computer Society Press.

[44] L.I. Lieberman and M.A. Wesley. AUTOPASS: An automatic programming system for computer controlled mechanical assembly. *IBM Journal of Research and Development*, 21(4):321–333, 1977.

[45] Y. Liu. A sensor-based robot planner for assembly tasks. In *First International Artificial Intelligence Conference in Japan*, Osaka, Japan, October/November 1987.

[46] Y. Liu. A justification for the characteristic invariant representation of TR subgroups of the proper Euclidean group. In *The fifth SIAM Conference on Discrete Mathematics*, Atlanta, Georgia, June 11-14 1990.

[47] Y. Liu and M. Arbib. A planner for sensor-based robotics in the assembly domain. Technical Report COINS TR86-36, Department of Computer and Information Science, University of Massachusetts, Amherst, MA., 1986.

[48] Y. Liu and R.J. Popplestone. Assembly planning from solid models. In *IEEE International Conference on Robotics and Automation*, pages 222–227, Washington, D.C., 1989. IEEE Computer Society Press.

[49] Y. Liu and R.J. Popplestone. Assembly feature mating inference from solid models using symmetry groups. In C.Y. Ho and G. Zobrist, editors, *Progress in Robotics and Intelligent Systems*. Ablex Publishing Corporation, New Jersey, 1990. To appear.

[50] Y. Liu and R.J. Popplestone. Symmetry constraint inference in assembly planning. In *Eighth National Conference on Artificial Intelligence*, Boston, Mass., July/August 1990.

[51] T. Lozano-Pérez. Task planning. In M. Brady et.al., editor, *Robot Motion: Planning and Control*. MIT Press, 1982. The MIT Press Series in Artificial Intelligence.

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

[53] T. Lozano-Pérez et.al. Handey: A robot system that recognizes, plans, and manipulates. In *IEEE International Conference on Robotics and Automation*, pages 843–849, Washington, D.C., March 1987. IEEE Computer Society Press.

[54] D.M. Lyons. Extended abstract: An algebra of robot task plans. In *IEEE Symposium on Foundations of Computer Science*, October 1988.

[55] S. Mac Lane and G. Birkhoff. *Algebra*. MacMillan Publishing Co., Inc., New York, second edition, 1979.

[56] A.K. Mackworth and E.C. Freuder. The complexity of some polynomial network consistency algorithms for constraint satisfaction problems. *Artificial Intelligence*, 25:65–74, 1985.

[57] M.T. Mason. Compliance and force control for computer controlled manipulators. In M. Brady et.al., editor, *Robot Motion: Planning and Control*. MIT Press, 1982. The MIT Press Series in Artificial Intelligence.

[58] W. Miller Jr. *Symmetry Groups and Their Applications*. Academic Press, New York, 1972.

[59] R. Mohr and T.C. Henderson. Arc and path consistency revisited. *Artificial Intelligence*, 28(2):225–233, 1986.

[60] E.E. Moise. *Geometric Topology in Dimensions 2 and 3*. Springer-Verlag, New York, 1977.

[61] J.L. Nevins and D.E. Whitney. Assembly research. *Automatica*, 16(6), 1980.

[62] R.P. Paul. *Robot Manipulators: Mathematics, Programming, and Control*. MIT Press, Cambridge, MA., 1981.

[63] R.J. Popplestone. Group theory and robotics. In M Brady and R. Paul, editors, *Robotics Research, The First Int. Symp.* MIT Press, Cambridge, MA., 1984.

[64] R.J. Popplestone. The Edinburgh Designer System as a framework for robotics: The design of behavior. *AI EDAM*, 1(1), 1988.

[65] R.J. Popplestone, A.P. Ambler, and I. Bellos. An interpreter for a language for describing assemblies. *Artificial Intelligence*, 14(1):79–107, 1980.

[66] R.J. Popplestone, R.A. Grupen, Y. Liu, G. Dakin, D. Oskard, and S. Nair. Planning for assembly with robot hands. In *SPIE Conference on Intelligent Robotic Systems*, Philadelphia, PA., November 1989.

[67] R.J. Popplestone, Y. Liu, and R. Weiss. A group theoretic approach to assembly planning. *Artificial Intelligence Magazine*, Spring 1990.

[68] R.J. Popplestone, R. Weiss, and Liu Y. Using characteristic invariants to infer new spatial relationships from old. In *IEEE International Conference on Robotics and Automation*, Philadelphia, PA., April 24-29 1988.

[69] E.G. Rees. *Notes on Geometry.* Springer-Verlag, New York, 1983.

[70] A.A.G. Requicha and Voelcker. Solid modeling: A historical summary and contemporary assessment. *IEEE Computer Graphics Applications*, 2(2):9–24, March 1982.

[71] A.A.G. Requicha and Voelcker. Solid modeling: Current status and research directions. *IEEE Computer Graphics Applications*, 3(7):25–37, October 1983.

[72] E.D. Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, 5:115–135, 1974.

[73] E.D. Sacerdoti. *A Structure for Plans and Behavior.* North-Holland, New York, 1977.

[74] A.C. Sanderson and L.S. Homem-de Mello. Task planning and control synthesis for flexible assembly systems. In *ATO Internaltional Advanced Research Workshop on Machine Intelligence and Knowledge Engineering for Robotic Applications*, Maratea, Italy, May 12-16 1986.

[75] D. Schattschneider. The plane symmetry groups: Their recognition and rotation. *American Mathematical Monthly*, 85:439–450, 1978.

[76] A.V. Shubnikov, N.V. Belov, and et.al. *Colored Symmetry.* Macmillan Company, New York, 1964. W.R. Holser, editor, translated from Russian by J. Itzkoff and J. Gollob.

[77] A.V. Shubnikov and V.A. Koptsik. *Symmetry in Science and Art.* Plenum Press, New York, 1974. D. Harker, editor, translated from Russian by G.D. Archard.

[78] A. Sloman and S. Hardy. POPLOG: A multi-purpose multi language program development environment. *AISB Quarterly*, 47, 1983.

[79] M. Stefik. Planning with constraints (MOLGEN: Part 1). *Artificial Intelligence*, 6:111–140, 1981.

[80] M. Stefik. Planning and meta-planning (MOLGEN: Part 2). *Artificial Intelligence*, 6:141–170, 1981.

[81] F. Thomas and C. Torras. A group-theorectic approach to the computation of symbolic part relations. *IEEE Journal of Robotics and Automation*, 4(6), December 1988.

[82] R. Waltzman. Finding symmetries of polyhedra. Technical Report CAR-TR-333, CS-TR-193, Center for Automation Research, University of Maryland, College Park, MD., 1987.

[83] D.E. Whitney et.al. Tools for strategic product design. In *NSF Engineering Design Research Conference*, pages 581–595, College of Engineering, University of Massachusetts, Amherst, MA., June 11-14 1989.

[84] J.D. Wolter. *On the Automatic Generation of Plans for Mechanical Assembly*. PhD thesis, University of Michigan, 1988.

[85] J.D. Wolter, T.C. Woo, and R.A. Volz. Optimal algorithms for symmetry detection in two and three dimensions. In *The Visual Computer*. Springer-Verlag, New York, 1985.

[86] T.C. Woo. Automatic disassembly and total ordering in three dimensions. In *Manufacturing Analysis and Synthesis, PED-Vol.25*, pages 291–303, Boston, MA., December 13-18 1987. Winter Annual Meeting of the American Society of Mechanical Engineers(ASME).

[87] R.F. Woodbury and I.J. Oppenheim. An approach to geometric reasoning in robotics. *IEEE Transactions on Aerospace and Electronic Systems*, 1988.

[88] A. Zahnd, S. Nair, and R.J. Popplestone. Tractable Subgroups of the Euclidean group. Technical Report COINS TR89-51, Department of Computer and Information Science, University of Massachusetts, Amherst, MA., March 1990.