ANALYSIS AND DEVELOPMENT OF A GENERIC GRIPPER FOR AUTOMATED PART RECOGNITION AND ASSEMBLY

By
JIANAN HUANG

A thesis presented as partial fulfilment of the requirements for the degree

DOCTOR OF ENGINEERING

in
MECHANICAL AND MANUFACTURING ENGINEERING

in the
Faculty of Engineering

at
RAND AFRIKAANS UNIVERSITY

SUPERVISOR: PROF. Z. KATZ

SEPTEMBER 1997
ABSTRACT

The grasping strategy for a three dimensional object by a robotic gripper requires a geometrical reasoning and analysis of the physical gripper design, control and operation. The work addresses the problem of data acquisition and processing required for an object recognition and its application in the selection of grasping strategy for a given gripper.

The system described in the thesis integrates the analyses of image data, object geometry and grasping operation in a systematic way. It is hierarchically constructed in several levels of analyses and processes including object recognition, grasping feature representation and classification, matching strategy for objects and the gripper and grasping description and operation. Object shape features are taken for recognition based on the image data collected through an infrared sensor. With a face relation graph proposed, an object model is built for describing the object geometrical properties and extracting its grasping features. A coding system based on group technology concepts is proposed for object classification. It describes object features relative to grasping operation. Gripping models are established and incorporated with the coding system for analysis of object gripping features. By means of the gripping models and the coding system, objects to be grasped are classified and grouped into specific families according to their similarities in gripping.

The information transformation between the object and the gripper is made through a matrix representation. An object matrix describes the selection of gripping faces and object geometry for gripping, while a gripper matrix describes the fingers selection and its configuration in correspondence with the object to be grasped. The matching of the matrices is established through a knowledge-based reasoning approach. The grasping operation is controlled by a computer in terms of the commands generated by the gripper matrix through a gripper code. The design of the generic gripper for this application is described.
ACKNOWLEDGEMENTS

I would like to express my deep appreciation to my wife Aihua Liang and my son Huang Huang as well as my parents and parents-in-law. Without their support, patience and understanding, I could not pursue my studies and complete the project.

I would like to express my sincere thanks to my supervisor Prof. Z. Katz for his interest, guidance and support throughout the project.

I would like to thank the staff and students of the system laboratory of the Engineering Faculty for their kind help in many aspects.
TABLE OF CONTENTS

ABSTRACT ii
ACKNOWLEDGEMENTS iii

CHAPTER 1 INTRODUCTION
1.1 Motivation of this Study 1
1.2 Purposes of this Study 4
1.3 Layout of this Thesis 4

CHAPTER 2 RELEVANT WORK
2.1 Introduction 6
2.2 Automatic Assembly 6
2.2.1 Description of Object 7
2.2.2 Mating Features and Spatial Relationships of Parts in Assembly 15
2.2.3 Assembly Task Planning 18
2.2.4 Design for Assembly 25
2.3 Application of Image Processing and Object Recognition 26
2.3.1 Image Processing 27
2.3.1.1 Binary Image and Thresholding 28
2.3.1.2 Image Smoothing 30
2.3.1.3 Edge Detection 33
2.3.2 Model Based Object Recognition 37
2.3.2.1 Feature Extraction 38
2.3.2.2 Object Modelling and Recognition 40
2.4 Group Technology and Its Application 41
2.4.1 Principle of Group Technology 42
2.4.2 Classification Methods 43
2.5 Robotic Grippers 47
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5.1</td>
<td>Grasp Modelling and Planning</td>
<td>47</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Sensor Based Grasping Analysis</td>
<td>48</td>
</tr>
<tr>
<td>2.5.3</td>
<td>grasping Force Analysis</td>
<td>49</td>
</tr>
<tr>
<td>2.5.4</td>
<td>Gripper Structure and Design</td>
<td>50</td>
</tr>
<tr>
<td>2.6</td>
<td>Conclusion</td>
<td>53</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>55</td>
</tr>
<tr>
<td>3.2</td>
<td>Outline of the Recognition System</td>
<td>56</td>
</tr>
<tr>
<td>3.3</td>
<td>Sensory Data Collection and Processing</td>
<td>57</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Sensor Structure and Data Collection</td>
<td>57</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Data Processing</td>
<td>59</td>
</tr>
<tr>
<td>3.3.2.1</td>
<td>Thresholding</td>
<td>60</td>
</tr>
<tr>
<td>3.3.2.2</td>
<td>Data Smoothing</td>
<td>62</td>
</tr>
<tr>
<td>3.3.2.3</td>
<td>Edge Detection</td>
<td>64</td>
</tr>
<tr>
<td>3.4</td>
<td>Object Features and Recognition</td>
<td>65</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Object Models and Shape Features</td>
<td>66</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Searching for Object Recognition</td>
<td>72</td>
</tr>
<tr>
<td>3.5</td>
<td>Experiment Setup</td>
<td>75</td>
</tr>
<tr>
<td>3.6</td>
<td>Conclusion</td>
<td>79</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>81</td>
</tr>
<tr>
<td>4.2</td>
<td>Object Gripping Models</td>
<td>82</td>
</tr>
<tr>
<td>4.3</td>
<td>Structure of the Coding System</td>
<td>85</td>
</tr>
<tr>
<td>4.4</td>
<td>Knowledge Based Code Generation</td>
<td>88</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Coding for Primary Features and Their Relations</td>
<td>92</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Coding for Face Types of Gripping</td>
<td>95</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Coding for Gripping Attributes</td>
<td>96</td>
</tr>
<tr>
<td>4.5</td>
<td>Conclusion</td>
<td>97</td>
</tr>
</tbody>
</table>
8.3 Program Implementation 137
8.4 Proposed Future Work 138

APPENDICES
APPENDIX A 140
APPENDIX B 142
APPENDIX C 144
APPENDIX D 145
APPENDIX E (Program disks)

REFERENCES 155

LIST OF TABLES
Table 4.1 Proposed coding system structure 86
Table 4.2 Coding digits for the primary features 87
Table 4.3 Coding digits for the face types and gripping attributes 88
Table 6.1 Proposed gripper code system 117

LIST OF FIGURES
Figure 1.1 Function flow of the proposed system 2
Figure 2.1 A CSG tree model 7
Figure 2.2 CSG primitives 8
Figure 2.3 A B-Rep Model 10
Figure 2.4 Inner and out loops 10
Figure 2.5 B-Rep model showing two different objects with same topology 11
Figure 2.6 The data structure to represent the boundary 16
Figure 2.7 Example of liaison diagram 19
Figure 2.8 All valid liaison sequences of the product in Figure 2.7 20
Figure 2.9 AND/OR graph for a product 23
Figure 2.10 The generic model of a vision system 26
Figure 2.11  An ideal grey-level histogram
Figure 2.12  3 x 3 convolution filter h
Figure 2.13  Gaussian curves with different $\sigma$
Figure 2.14  Convolution masks: (a) Roberts; (b) Sobel; and (c) Prewitt edge detection operators
Figure 2.15  A general paradigm in model-based vision
Figure 2.16  Features for various shapes
Figure 2.17  Parts similarity in manufacturing
Figure 2.18  Chain-type code
Figure 2.19  Hierarchical coding
Figure 2.20  Hybrid coding structure
Figure 2.21  Basic structure of the Opitz coding system
Figure 2.22  Two- and three-fingered grippers
Figure 2.23  Dextrous robot hands
Figure 3.1   System schematic of model-based object recognition
Figure 3.2   The structure of an infrared sensor
Figure 3.3   Sensory data of a block
Figure 3.4   Image histogram of a block
Figure 3.5   Binary image by thresholding
Figure 3.6   3 x 3 neighbourhood average mask
Figure 3.7   Image by smoothing
Figure 3.8   Image data by edge detection
Figure 3.9   Face classification
Figure 3.10  Object view
Figure 3.11  Object modelling with three levels
Figure 3.12  Face relation graph
Figure 3.13  The recognition searching
Figure 3.14  Experiment setup for object recognition
Figure 3.15  Sensor structure prototype
Figure 3.16  Sensory plane and coordinates
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.17</td>
<td>Circuits of the emitter and the receiver</td>
</tr>
<tr>
<td>4.1</td>
<td>Gripping model construction</td>
</tr>
<tr>
<td>4.2</td>
<td>Knowledge based system for generation of digits</td>
</tr>
<tr>
<td>4.3</td>
<td>The precedence of filters</td>
</tr>
<tr>
<td>4.4</td>
<td>Filtering procedure</td>
</tr>
<tr>
<td>5.1</td>
<td>Typical form of the object matrix</td>
</tr>
<tr>
<td>5.2</td>
<td>Transformation of object code (model) and matrix</td>
</tr>
<tr>
<td>5.3</td>
<td>Typical form of the gripper configuration matrix</td>
</tr>
<tr>
<td>5.4</td>
<td>Information flow and matching strategy</td>
</tr>
<tr>
<td>5.5</td>
<td>Matching stages</td>
</tr>
<tr>
<td>5.6</td>
<td>Face-finger pairs search tree</td>
</tr>
<tr>
<td>5.7</td>
<td>Knowledge base consisting of four levels</td>
</tr>
<tr>
<td>6.1</td>
<td>Schematic for generating operation commands</td>
</tr>
<tr>
<td>6.2</td>
<td>Coordinate systems and geometrical transformation</td>
</tr>
<tr>
<td>6.3</td>
<td>Control schematic</td>
</tr>
<tr>
<td>6.4</td>
<td>Motor control circuit</td>
</tr>
<tr>
<td>6.5</td>
<td>Experiment setup of the gripper control</td>
</tr>
<tr>
<td>7.1</td>
<td>The gripper with three fingers</td>
</tr>
<tr>
<td>7.2</td>
<td>A finger structure detailed design</td>
</tr>
<tr>
<td>7.3</td>
<td>Three finger position arrangement</td>
</tr>
<tr>
<td>7.4</td>
<td>Force diagram of joint A</td>
</tr>
<tr>
<td>7.5</td>
<td>Gear drives in joint B</td>
</tr>
<tr>
<td>7.6</td>
<td>Force diagram of joint C</td>
</tr>
<tr>
<td>A.1</td>
<td>The distance $a_{ij}$ measurement</td>
</tr>
</tbody>
</table>
CHAPTER 1 INTRODUCTION

1.1 Motivation of this Study

In order to accomplish an assembly task in robotic environment, design of robotic grippers, choice of gripping models, feature analysis of an object to be grasped and information transformation between the object and the gripper are important matters which, even though studied by many investigators, still have many problems to be further investigated and solved. Up to now, most studies of gripping focused on the geometric analysis using CAD models in the task level or were based on the analysis of human hands prehension, and separately analysed object features and the control of grasping operations. A little work has been done for easy and effective transformation of information from an object to a gripper and for integration of an image analysis, object recognition, grasping feature extraction and operation command generation as well as grasping control in a systematic way.

From the stand of manufacturing and assembly, this study proposes a coding system for representation of gripping and a matrix description for transformation of grasping information. An integrated system is built for analysis of object features, gripping performance and for establishment of the correspondence and transformation between an object and a gripper. The system is grasping oriented.

Since mechanical parts are of multitypes with different shapes and some of them have specific functional requirements on assembly, the assembly of these parts automatically becomes more complicated and grasping these parts properly to carry out the assembly process becomes more important. Although these parts have different shapes, any of them, even it has a complicated shape, can be regarded as a combination of some basic objects (or shapes) such as block, polyhedron, cylinder and so on. The grasping strategy made in the thesis is based on these basic objects.
The function flow of this system proposed in this study is illustrated in Figure 1.1.

Figure 1.1 Function flow of the proposed system
When an object (or a part) is properly fed into the assembly site and a gripper starts to grasp the object, there are several problems needed to be solved. It should be known the type of the object, its location and orientation, which parts (or faces) of the object to be touched by the fingers of the gripper, how to transfer the object's information to the gripper and how to control the gripper.

In the study, the image data of an object on the scene is obtained through a simple infrared sensor and the object is recognised two-dimensionally through the analysis of the image data. Two models, namely object model and object gripping model, are built in the system. The object model is used for description of the object geometrical features for object recognition, while the object gripping model is for representation of the object primary gripping features. As an object is recognised based on its object model, its gripping model is extracted from the description of its geometry and function with the object model.

Based on group technology, the coding system proposed in the study is used to describe object features relative to grasping analyses and operations. Accordingly an object to be grasped is classified in terms of its gripping model. The grasping features and parameters of the object are encoded by means of the coding system corresponding to its gripping model.

The information transformation between the object and the gripper is made through a matrix representation. An object matrix describes the selection of gripping faces and object geometry for gripping, while a gripper matrix describes the fingers selection and its configuration in correspondence with the object to be grasped. The matrices are generated in terms of the object code and the information from its gripping model. The matching of the two matrices is established through a knowledge-based reasoning approach. After the matching, the gripping information of the object is transferred to the gripper and simultaneously a gripper configuration is formed with respect to the object. A gripper code for detailed descriptions of the joints and joints motions of fingers is
extracted from the gripper matrix and a series of operation commands are represented with the digits of the gripper code. The grasping operation is controlled by a computer in terms of the commands based on the gripper matrix and the gripper code. Part of the studies has been published in relevant international conference and journal [111] [118].

1.2 Purposes of this Study

The purposes of this research project are:

1. To provide the background of the research development of the gripping strategy and the gripping analysis and operation.
2. To provide an object based recognition system to recognise objects on the scene.
3. To illustrate the structure and function of the coding system and the gripping model for description of the object gripping features and for classification of those objects to be grasped.
4. To describe the matrix structure for information transformation between an object and a gripper.
5. To study the knowledge-based system for grasping analysis and reasoning.
6. To design a specific gripper with three fingers for application and demonstration in the project.
7. To build an experiment set-up for demonstration for the project.

1.3 Layout of this Thesis

Chapter 1 is the general introduction of the thesis. Chapter 2 gives a literature overview related to the study. This chapter emphasises on the approaches to object representation, the techniques of image processing and object recognition, and the grasping analysis and planning.
Chapter 3 illustrates a model-based object recognition system for the study. The emphasis of this chapter is placed on the object shape features, the feature description and extraction, and the object model representation.

Chapter 4 describes the coding system for representation of the grasping features of objects based on group technology. The emphasis of the chapter is on gripping models and knowledge-based coding digits generation.

Chapter 5 discusses the matrix representation for establishment of the correspondence of the object and the gripper. This chapter emphasises on the matching of the gripper and the objet matrices, and knowledge representation and reasoning in general and in this regard.

Chapter 6 describes the gripper code, operation commands and the gripper control strategy. The emphasis of the chapter is on representation of the joint motion of the gripper.

Chapter 7 illustrates the structural design and the force transmission mechanism of the gripper. Both of the gripper structure and force transmission are emphasised in the chapter.

Chapter 8 gives general conclusions of the thesis. In addition, the program implementation of the system and the proposed future work in this research are also discussed in the chapter.
CHAPTER 2 RELEVANT WORK

2.1 Introduction

Assembly plays an important role in manufacturing. In many companies, assembly costs have covered more than half of the total costs in manufacturing of a product [1]. Assembly operations are regarded as an field with big potential for robot applications [2]. In the basis of economics and the technological capabilities of robots, the need of robots for automatic assembly in batch production is increased fast. At the beginning of 1987, there were 27,600 robots installed in the United State, 5,450 (19.7%) of which were applied in assembly. By 1991 the application of robots for assembly had been increased almost 30% [1].

The development of automatic assembly with robotic grippers requires a wide range of knowledge and technology in assembly planning, sensors, image processing and object recognition, gripping analysis and reasoning, gripper design and control, and so on. Many investigations and researches have been contributed to these aspects. This chapter provides a summary of these previous work done by researchers. These aspects to be reviewed in this chapter include the methodology of automatic assembly, the theory and applications of image processing and object recognition, the principle and applications of group technology and the gripper design and control.

2.2 Automatic Assembly

Automatic assembly requires information regarding the product design and representation and the assembly tasks as well as the process information on the assembly sequence, the robot motion and trajectory planning and the gripper design and grasp planning. The robot motion and trajectory planning have no close relations with my project, so they are neglected in the review. The gripper design and grasp planning are reviewed in the section of robotic grippers.
2.2.1 Description of Objects

Computer-Aided Design (CAD) provides a means to represent objects for manufacturing. CAD can be defined as the use of computer systems to assist in the creation, modification, analysis, or optimisation of a design [3] [4]. For assembly of a product, every part (object) of the product is already designed and manufactured. Thus the analysis of these objects using a computer is a main task in assembly. There are three major methods for representation of an object with the computer, namely Constructive Solid Geometry (CSG), Boundary Representation (B-Rep) and feature based design.

A CSG model uses a binary tree to represent an object. The binary tree consists of geometric primitives, transformations and symbols representing Boolean operators. An example is shown in Figure 2.1.

![Figure 2.1 A CSG tree model [4]](image-url)
In the CSG tree, terminal nodes describe either primitives or primitives with transformation and nonterminal nodes describe either Boolean operators or transformation. An object model can be completely established by these tree building elements, with topological information implied.

The primitives used in a CSG tree are a set of pre-defined primitives in the CAD systems or created by the users. In a solid modelling, block, cylinder, sphere, torus, wedge, and cone (Figure 2.2) are commonly used primitives.

Parameters are given for each primitive to define its size. For example, the parameters of length, width and height are used for definition of a block; the radius and length are for a cylinder. These primitives are further defined internally by their bounding faces, edges and vertices.

Figure 2.2 CSG primitives [4]
The Boolean operators used in the CSG construction are union (\(\cup\)), difference (\(-\)), and intersect (\(\cap\)). Applying these operators to the object in Figure 2.1, the object can be expressed as:

\[
O_7 = P_4T_4 - *(P_1T_1 \cup P_2T_2)T_5 - P_6T_6
\]

The same object can also be represented as:

\[
O_7 = P_4T_4 - *(P_1T_1T_5 \cup P_6T_6) - *P_1T_1T_5
\]

Where \(T_i\) is a transformation matrix including rotation, translation and scaling operations. The \(T_i\) is used for positioning and orienting a primitive or an intermediate object model. \(P_i\) indicates the positions of these primitives. There are other combinations to built the CSG trees for the object using the Boolean operators [4]. Using CSG models, obviously the relationship of each primitive of an object can be logically represented. However, since a CSG model does not represent face, edge and vertex information explicitly, it is not possible to make an interactive editing for the local geometry of the object and not easy to directly display the object model either. Furthermore, it can not directly provide face and edge information for manufacturing.

B-Rep representation is a boundary model. Using this model, an object is represented by segmenting its boundary into a finite number of bounding faces. And each face is represented by its boundary edges (loops) and vertices as described by Requicha [5]. The same object as Figure 2.1 can be described as a B-Rep model shown in Figure 2.3.
A loop is an orderly list of edges and its direction may be decided by the face vector pointing to the inside of the solid using right hand rule. Therefore, the inside and outside of the object can be distinguished on a same face. For example, there are three loops on a face shown in Figure 2.4, an outer loop of straight edges and two inner loops of a circle and straight edges. One inner loop describes a protrusion and the other represents a depression.

Basically, boundary model is a topologically explicit representation. The B-Rep tree is a topological relation tree. In a B-Rep model tree, the nodes of the tree are faces, edges and vertices. The terminals of the tree are geometrical entities and normally vertices.
There are no operators in the tree. In Figure 2.5, the topological relations of faces, edges and vertices for part A and part B are completely identical. For the two objects, the only difference between them is the coordinates of the points in their B-Rep trees.

![Part A and Part B](image)

**Figure 2.5** B-Rep model showing two different objects with same topology [6]

Based on explicitly representing faces, edges and vertices in boundary representation, the detailed geometry of an object model can be easily described and modified, and the design data (face and edge definition) be again used in NC application in manufacturing. However, a B-Rep model is usually larger than the corresponding CSG model. Thus it needs more memory to store in a computer. The model is also not unique. Furthermore, for the higher-level features of an object, it needs further to describe the features from its face-edge-vertex data model.

Feature based representation (or design) is widely used in computer-aided design and manufacturing. Feature based design means a design with features. However, there does not exist an agreed-on definition for the item of feature. Definitely feature is application
specific. For example, the features of objects in assembly largely imply how these objects are assembled together. A definition of feature from Nnaji [6] is that a feature is 'a set of surfaces together with specifications of the bounding relationships between them and which imply an engineering function (or stereotypical entity) on an object and which may be formed on the faces, edges, or corners of an object'. This definition captures both geometrical and functional implications of a feature.

Usually, features of mechanical parts may be catalogued into form features and functional ones according to their geometrical and functional characteristics respectively. In most applications, form features are much been concerned, with functional characteristics implied. Chang [4], based on the geometry, classifies features into the following:

- **Face features** - those defined by two or three dimensional faces. The face features may be used to 2D or surface modelling CAD systems.
- **Volumetric features** - those defined by three dimensional and enclosed volumes. The volumetric features are better used with a CSG modelling tree to construct a solid model.

Based on the applications:

- **Design features** - those meaningful to design. The design features, such as hole, chamfer, groove, countersink, counterbore, fillet, etc., are used in engineering design.
- **Manufacturing features** - those meaningful to manufacturing. Actually, manufacturing features are identical with or similar to design features. Both design and manufacturing features may either be face features or be volumetric features. Based on CAD data and design features, manufacturing process planning, assembly analysis and sequence planning, CAD and NC (numerical control) integration can be made. Eventually, CAD/CAM integration and computer-integrated manufacturing (CIM) can be achieved [6].
A CSG tree, B-Rep modelling and feature based representation are widely used in mechanical design, manufacturing, robotic assembly and so on. Feature based representation has several advantages over the conventional CSG and B-Rep approaches as follows [7].

- Providing the ability for capture of designer's intent and knowledge in a variety of engineering applications.
- Using a more flexible and intelligent way to build a solid model than the CSG binary tree.
- Providing a means for designers to work with high-level features instead of low-level primitives in design and analysis.
- Better understanding a part geometry and its functionalities.
- Encouraging towards standardisation in order for improvement of manufacturability and product quality.

Since a conventional CSG or B-Rep model can only handle basic shape data of an object and are not properly manufacturing-oriented, Roy and Liu [8] proposed a feature-based representation scheme based on the hybrid CSG/B-Rep data structure. The hybrid structure takes the advantages of both CSG and B-Rep models and is coupled with a structured face-adjacency graph representation (SFAG), not only for description of geometry but also for representation of dimensioning and tolerancing information. Shah et al [9] developed a system called FBMS consisting of an advanced solid modelling shell and a feature mapping shell. The functional requirements for an integrated system were discussed and the concept design of feature based modelling was presented in the paper. In terms of CAD/CAM integration and production engineering, features in the system are divided into form features for geometry, material features for material properties and treatment, precision features for tolerances and surface finish, and technological features for production performance. Production information such as automatic GT classification and process selection for machining can be provided by the system through feature modelling and mapping.
Nitschke et al [10] illustrated a feature extraction interface in a framework which is used to construct a feature based part model. The form features of a part are obtained from a non-feature based CAD database in the procedure of feature extraction. Features are extracted in the sequence of primary features (parents) and then secondary features (their children). The dimensions, positions and their spatial relations of each feature are extracted and organised into a feature graph. Then reconstruction of a model is made by means of this feature graph and the non-geometrical attributes such as material, function and precision can be specified with construction of the model, since these data are stored as part of the feature based models. Henderson and Anderson [11] studied feature extraction using logic programming. The approach developed by them can automatically extract manufacturing information in the form of part features. In the process a part description is searched, its cavity features are recognised by logic reasoning using production rules, and then the features are extracted and organised into a feature graph which shows feature definition, feature adjacency and feature accessibility describing the path from the stock material to the feature of interest.

Using feature based representation can support the reasoning about the topology and geometry of an object as well as its manufacturing activities in developing intelligent CAD systems. Cunningham and Dixon [12] investigated the origin of features, and their role in an intelligent design with features system. Representations in terms of features are needed for knowledge based systems used in various design and manufacturing applications. Different manufacturing processes requires different features for their various descriptions and activities. Features are identified with respect to the heuristic knowledge in a manufacturing process. Examples of features for reasoning about die design, process planning and inspectability were given by Cunningham and Dixon in their paper [12]. Shah and Roger [13] applied the artificial intelligence techniques to create feature models. The inheritance rules were used to establish the relationships among features through a frame-based data structure, while the cognition rules were to place constraints on these features. In their investigation, an expert system shell in terms of the
knowledge based feature modelling was built to extract feature information from the design data for manufacturing applications. In the framework of feature based part modelling developed by Chen et al [7], a feature based modelling environment integrated with a knowledge based environment was described in order to increase the capabilities and intelligence of CAD/CAM systems. The data of a part model are extracted from the part database created by feature based design, and the part model is constructed in the knowledge based environment of manufacturing. Therefore the high-level part model can support many tasks requiring geometrical reasoning, such as manufacturability assessment, assemblability assessment, and cost estimation. Feature reasoning for automatic robotic assembly was investigated in detail by Nnaji and Liu [14], which is reviewed in the next section.

2.2.2 Mating Features and Spatial Relationships of Parts in Assembly

In assembly, when two parts are assembled together, there exits an area where the two parts have common boundaries overlapped. This area may be divided into a number of faces of each part. These faces constitute the mating features with respect to each part. In order to make an assembly planning, it is required to represent and reason about objects geometrically and functionally, in which an important process is to describe and identify the mating features of these objects.

Nnaji and Liu [14] studied these issues mentioned above in detail. In the system they developed, the knowledge of the workspace and the features of objects within the workspace are represented, and the process of reasoning about the object shapes and their behaviour in assembly situation is then carried out. And therefore, the compatible sets of mating features can be identified in the reasoning process for further assembly planning. The representation of objects in the system is made in a hybrid format based on CSG trees, where the nodes of the CSG trees are represented in a LISP-like format and the primitives in a PROLOG format. For example, the union and intersection of shapes S1 and S2 are represented as [union, S1, S2], and [intersect, S1, S2], and the
difference as \([diff, S1, S2]\). The shape \(S\) of an object \(C_n\) is described in the PROLOG database as \(csg(C_n, S)\). The geometrical entities of vertex, edge, loop, plane and face are used for boundary description (refer to Figure 2.6) which form the polyhedral representation of the body (planes, faces etc.).

The generic primitive database in the system is used for store of those primitives with all the dimensions equal to 1 unit. For example, a generic cube is a cube with its length, width and height equal to 1 unit and a generic cylinder is a cylinder that has a 1 unit radius and height. These generic primitives are used for matching the primitives of an object, through scaling and transformation steps in a coordinate frame. In the system, the concave features of objects are taken as major features for mating. If two shapes can be mated in assembly, it is likely that one shape has a convex feature and the other shape has a concave feature. Therefore an object with a concave shape may act as the possible mating place for an object that has a convex shape. The function of feature reasoning is to find the possible mating place for assembly. Nnaji and Liu used the interpretation rules to represent these concave and convex features. These rules are easily coupled with the
reasoning process. Furthermore feature relationships and feature trees are built using the PROLOG clause. For applications, the approach developed by Nnaji and Liu for reasoning about features in a systematic way can be applied to reasoning about the mating features, find the feasible grasping sites, and generate NC programs for machining operations.

When two parts are assembled together, the spatial relationships between them strongly affects on the assembly operation and grasping strategy. In order to specify the spatial relationships, Amber and Popplestone [15] described an approach to express the relative positions of parts in some assembly state (goal state) of a product by specifying the spatial relationships of features. The features for describing bodies used by them are a plane face, a cylindrical shaft and a cylindrical hole. Two spatial relationships, namely against and fits, were used to describe the interactions between features of bodies. Moreover a system called RAPT [16] was developed by Popplestone et al for inference of the positions of bodies from specified symbolic spatial relationships between features of bodies. Popplestone et al [17] extended their work in ref. [15] by adding some new body features - an edge, a vertex and a spherical face, and a new spatial relationship of coplanar. Thus the against feature may be face to face, or face to shaft, or edge to face, or spherical face to face, or vertex to face, or vertex to vertex. The fits feature indicates that a shaft fits a hole when their X axes lie along the same line, but in opposition. And the relationship of coplanar is defined as that a face is coplanar with another face when they lie in the same plane with their X axes in the same direction. In addition to describing states they in this paper [17] described actions which transform one state to another. The action descriptions act as inferences to reason about the positions of bodies in terms of their spatial relationships.

Lee et al [18] [19] proposed a hierarchical data structure for representing assemblies. The data structure is a tree structure using the concept of virtual link to represent the relationships between assembly parts. A virtual link is defined as the set of information required to describe the assembly relationships e.g. rigid attachment, conditional
attachment, translational constraint and rotational constraint, and the mating features between a mating pair. In terms of the virtual links, assembly data can be represented hierarchically, and the transformation matrix for describing the position and orientation of parts and subassemblies need not be assigned, since it can be derived from the mating feature information created by the virtual links. The mating features given by Lee et al are against and fits which are corresponding to the spatial relationships proposed by Popplestone et al. Using these mating features Rocheleau and Lee [20] described a system for interactive assembly modelling, in which the spatial relationships of parts are modelled by equations (assembly equations) derived from the mating conditions. A modified Newton-Raphson iteration incorporating a least squares technique is applied in the solution to the set of assembly equations.

Nnaji [21] has modified and expanded the concept of spatial relationships. The design with spatial relationships was defined in their system used not only for inferring the assembly position of parts but also as a model to capture the designers' intent. Six types of spatial relationships for assembly were defined by them, which are against, parallel-offset, parax-offset, aligned, incline-offset and include-angle. Among the six types the against relationships are the most basic spatial relationships which represent two faces physically touching one another at some point and widely used for description of assembly. A rule based system was developed in their system for selection of assembly features in terms of their spatial relationships and for inference of the final state of degrees of freedom.

2.2.3 Assembly Task Planning

In terms of assembly tasks, assembly planning deals with the geometrical and functional features and relations between assembled parts in order to generate feasible assembly plans and provide better assembly sequences to meet assembly efficiency and cost-effective requirements. In robotic assembly, more factors such as grasping, robot motion need to consider.
Algorithms based on the liaison diagram and precedence relationships of mating parts, through the process of answering yes-no questions by the designers, to generate all assembly sequences were proposed and developed by Bourjault, De Fazio and Whitney [22]. A liaison diagram is a network wherein nodes represent parts and lines between nodes represent the relations between parts. The type of relations between parts may be defined by users. Figure 2.7 is an example of a liaison diagram to describe the assembly relations of parts for the product. All valid liaison sequences (assembly sequences) for the product in Figure 2.7 are displayed by a graphical representation shown in Figure 2.8.

Figure 2.7 Example of liaison diagram [22]
Based on the algorithms mentioned above, Baldwin et al [23] developed an integrated set of user-interactive computer programs used for generating all feasible assembly sequences for a product, and for aiding the users in evaluating and selecting the assembly sequences. In the programs a disassembly procedure is taken for analysing and
generating the sequences, and the on-line visual aids are provided during the processes of
generation and evaluation. In evaluation matters of assembly difficulty, stability,
fixturing, orientation etc. are considered to highlight the desirable or undesirable
sequences. With the programs users may evaluate and select assembly plans in the early
stage of design. Lee [24] proposed a method for automatically generating assembly
sequences based on a liaison graph representation. An attributed liaison graph with
frames attached to individual nodes and edges of the liaison graph is built to describe
parts relations. The attributes of a liaison include the geometrical features of mating parts
and the interconnection mechanism between parts. The attributed liaison graph is further
transformed into an abstract liaison graph by merging those parts that can not be
mutually separable at the current stage of planning due to interconnection infeasibility
and functional dependency. In terms of the abstract liaison graph, a hierarchical
disassembly plan is then generated through decomposing assembly into subassemblies
recursively and selecting subassemblies based on the subassembly selection indices
values. Zussman et al [25] used a relational graph which links assembly features for
automatic assembly planning of 3D structures. The relational graph is similar to the
liaison graph, but the only difference between them is the definition and description of
vertices (nodes) and edges (lines) of graphs for assembly. The relational graph represents
object geometrical features and assembly features as well as their relations. For assembly
planning, an object is hierarchically represented by an object oriented approach within an
object model. The object relational graph is made with respect to the object model
representation. An algorithm was proposed for matching assembly features, i.e. to find
the mating features between the assembled objects. Subassemblies are deduced from the
relational graph by grouping objects together in terms of their spatial relationships.
Heuristic rules are used in the grouping. Sekiguchi et al [26] used a code to describe the
connective relations of a pair of parts. A matrix is built to express the assembly relations
of a product based on the code of the mating parts. Dini and Santochi [27] expressed an
assembly also using a matrix representation. Three matrices called the interference
matrix, the contact matrix and the connection matrix were built to analyse the assembly
and generate the assembly sequences.
Weule and Friedmann [28] developed a computer system of KOMPASS for computer-aided analysis in assembly planning of products. This system transfers data from a 3D CAD system which completely represents an assembled product geometrically. After the geometrical analysis the technological analysis is processed in terms of the predetermined assembly techniques of parts. Then the simulation of disassembly of parts is performed through a collisions-check, and the assembly precedence graph is interactively generated thereafter. These processes are displayed on a graphics monitor for user's check and approval. A flexible computer-aided process planning for assembly was proposed by Heemskerk and Reijers [29]. The assembly analysis is decomposed into four levels of abstraction: a batch level, a product level, a part level, and a primitive level. The batch level planning concentrates on decisions for a whole batch, which is closely related to the system layout design. The concept of sub-batch strategy is applied for single workstation batch assembly in the level. The product level makes decisions for a single product. Two steps are taken in the level. Firstly, rough sequences plans are generated directly from the descriptions of a product. Secondly, unstable subassemblies are discarded base on the analysis of assembly tasks and characteristics. And then the stable subassemblies are continuously decomposed into parts and actions. The part level and primitive level deal with each part and each primitive action in assembly. The low level action primitives are device driver-like procedures linked to real sensors and actuators. During planning, the assembly processes are simulated by means of these primitives; during plan execution, the real assembly actions are performed by physical devices through selection of these primitives. The best assembly plan is made according to the criteria of short assembly cycle time and plan robustness.

Homem de Mello and Sanderson [30] [31] used AND/OR graphs to represent all assembly plans explicitly. The graph is generated by decomposing all possible assemblies into subassemblies. And an assembly plan is generated by means of a heuristic search with respect to some evaluation functions. Generally the assembly problem can be converted into the reverse of solving disassembly problem. The AND/OR graphs, or called hypergraphs, are useful in representing disassembly problem. The nodes in the
The applications of the AND/OR graphs in assembly planning may be seen in [30-33]. A heuristic search for the best assembly plan in terms of two criteria for the evaluation and selection of assembly plans was introduced by Homem de Mello and Sanderson [34]. The two criteria are to maximise the number of different sequences encompassed by the assembly plan, and to maximise the amount of parallelism. The heuristic search is made over the AND/OR graph representation of assembly. Homem de
Mello and Sanderson [35] summarised and analysed five representations of mechanical assembly sequences. These representation methodologies are based on directed graphs, on AND/OR graphs, on establishment conditions, and on precedence relationships. The later includes two types: precedence relationships between the establishment of one connection and the establishment of another connection between parts, and precedence relationships between the establishment of one connection and states of the assembly process. The relationships between these five representations were analysed in their paper.

Artificial Intelligence is widely applied in robotic assembly planning. Fikes et al [36-38], Tangwongsan and Fu [39], and Fahlman [40] are some of the early investigators who studied the robotic planning problems using artificial intelligence methodologies and techniques. Fikes et al described a robot problem-solving system STRIP to perform robot tasks. The method of using a triangle table for planning, learning and executing were proposed. Tangwongsan and Fu proposed an approach of applying supervised learning to robotic learning. The knowledge of robotic tasks can be accumulated through the process of learning. The philosophy of the learning scheme is to use an analogy between a current unplanned task and any known similar tasks to reduce the searching speed. Fahlman described a system BUILD for planning for robot construction tasks. In the system a powerful heuristic control structure is built to support the planning. Using heuristics for assembly planning is also used over AND/OR graphs to find a best plan [30-33]. Kroll et al [41] proposed a knowledge based technique used for assembly planning. The theoretical and heuristic knowledge of mechanical components and assembly processes is represented by IF_THEN rules. The assembly sequence of a product is generated from the exploded-view of the product in terms of the knowledge rules. Tierney et al [42] applied an expert system to planning for robot-based flexible assembly systems. In the system new knowledge of robot-based assembly can be revised and the sample assembly plans are suggested for different products. Tonshoff et al [43] developed a knowledge-based system for automated assembly planning. In their system the assembly tasks are described by the jointing
positions between parts and by the assembly technology to be applied. The task-related rules are generated in a knowledge base for determining the assembly sequences.

2.2.4 Design for Assembly

Design for assembly (DFA) is a methodology which is taken to mean the design of a product for ease of assembly [44]. The objective of DFA is to integrate the product design and process planning into one common activity. As a central element of DFM (design for manufacture), DFA has one important characteristic of simplifying the product structure. Therefore based on the product structure simplification, the assembly cost and the total parts cost can be reduced and the product life cycle be shorted.

DFA analysis is taken in a systematic way. Some issues in automatic assembly of a product, such as part geometry and function, tool holder structures and properties, grasping and assembling planning, should be taken into account in the design stage. A CAD system can assist designers to set up a database of parts, to analyse the product structure, to carry out graphical assembly simulation, to evaluate assembly plans and to estimate part assemblability and assembly cost. On the other hand, it is necessary to incorporate product manufacturing data into the DFA system so that part manufacturing difficulties may be evaluated in the design cycle. With DFA analysis, products designed are ease for assembly and manufacture and should be more competitive in market.

A number of different DFA methods have been developed. They mainly include the Hitachi Assemblability Evaluation Method, the Lucas DFA Evaluation Method and the Boothroyd-Dewhurst DFA Method. For details of these methods, please refer to [44] and [45]. Knowledge based approaches have been employed in DFA analysis [45].
2.3 Application of Image Processing and Object Recognition

In sensor-based robotic assembly systems, the scene information is collected by visions or sensors. A generic model of a machine vision system can be defined and illustrated as in Figure 2.10.

Figure 2.10 The generic model of a vision system [46]

The scene refers to the assembly environment in which the assembled parts are to be fed in by a conveyer or other feeders, and the vision or sensor equipment is to be placed. The scene constraints put a priori constraints to the image system, such as the region the sensory system can cover and the reference points of the region for a robot gripper, in order to reduce the complexity of the system to a manageable level. The image acquisition deals with the process of transforming the light strength to a corresponding digital value. In our system, the electronic voltage relative to a light strength is transformed to a digital value through a A/D converter. Typically, there is 512 x 512 pixels resolution with each pixel representing a binary, grey scale or colour value in
terms of a vision system. The processes of contrast enhancement and adjustment, filtering to remove noise and improve quality, and correction for sensor distortion may be desirable in the stage of preprocessing. To some extent, the segmentation such as thresholding, edge detection may be included in the process of preprocessing. The preprocessing and segmentation is the initial stage for any subsequent recognition process. The feature extraction is an important process for recognition and classification of objects on the scene. It extracts the object size and shape information from the image data of an object. The classification is concerned with the process of pattern recognition or image classification. In the process some or all of the object features are utilised for making a decision which class or category of objects the unknown object belongs to. Finally, in terms of the information obtained from the previous steps the actuation process will be taken place. In robotic assembly systems the robots will be actuated to perform the assembly tasks during the process. In the following the techniques of image processing and object recognition are introduced briefly.

2.3.1 Image Processing

There are two requirements should be met by a method for representation of image data [46]: (1) the method should facilitate convenient and efficient processing by means of a computer; (2) it should encapsulate all information that defines the relevant characteristics of the image. Usually a conventional optical system delivers a continuous two-dimensional function, \( f(x, y) \) whose value represents the intensity of the light at that point. In terms of the requirements above, it is necessary to quantise the continuous function so that the function can be represented by an array of numbers and the image array can be processed with a digital computer.

In order for computer processing, as mentioned above, an image function \( f(x, y) \) must be digitised both spatially and in amplitude. The spatial digitisation at the spatial coordinates \( f(x, y) \) is referred to as image sampling, while the amplitude digitisation is called grey-level quantisation. In the spatial digitisation an image is sampled at \( m \times n \) discrete points.
Each sample is called a ‘picture cell’, ‘pixel’ for short. Here \( m \) and \( n \) are integers, and normally there has \( m = n \), and \( m \) and \( n \) are integral powers of 2. In the amplitude digitisation each pixel is assigned a numerical code which represents the intensity of the image at that point. The numerical code represents the resolution of each image point, which is determined by the number of quantisation levels, i.e. grey levels which are available between the extremes of intensity, black and white. The full range of the grey levels from black to white is referred to as a grey scale. The number \( G \) of grey levels is also chosen to be integral powers of 2 in the grey scale, i.e. \( G = 2^l \), where \( l \) is an integer. Generally, in order to encapsulate more information of an image the values of \( m \), \( n \) and \( l \) should be as high as necessary. However, in this situation more computation time will be needed and the processing efficiency be decreased. Therefore, for a specific application it is desired to choose proper values of \( m \), \( n \) and \( l \).

There are a great number of methods are applicable for image processing. Here only those methods relative to my applications are discussed.

2.3.1.1 Binary Image and Thresholding

In many applications, binary images are enough to represent the information desired, where an image is generated with only two grey levels, black and white. The binary image is very simple to store and manipulate with a computer as each pixel is represented by a single bit. For a more general grey scale image the binary image can be generated by applying an appropriate thresholding level to partition the general image into pixels with one of just two values. Hence, if

\[
f(x, y) = \text{intensity value at coordinates } (x, y) \text{ in the image} \\
g(x, y) \text{ denotes the binarised version of } f(x, y) \\
T = \text{thresholding value}
\]

the image represented by \( f(x, y) \) is binarised by applying the rules:
\[
\text{IF } f(x, y) \geq T \quad \text{THEN} \quad g(x, y) = 1 \\
\text{IF } f(x, y) < T \quad \text{THEN} \quad g(x, y) = 0. \quad (2.3)
\]

Obviously, the process of thresholding will assign a value of 0 to those pixels with a grey-level less than the thresholding level $T$ and a value of 1 to those pixels with a grey-level greater than the thresholding level. Thus the image is segmented into two disjoint regions, the black region (value 0) corresponding to the background, and the white region (value 1) to the object. More generally, a threshold operation may be regarded as a test involving some function of the grey-level at a point, some local (neighbourhood) property of the point, and the position of the point in the image. It is assumed that the function $T$ is given as

\[
T(f(x, y)) \quad (2.4)
\]

where $f(x, y)$, as beforementioned, is the grey-level at the point $(x, y)$ and $N(x, y)$ denotes some local property of the point $(x, y)$. Be placed restrictions on this function, three classes of thresholding, namely global, local and dynamic ones may be distinguished as follows:

(a) $T = T(f(x, y))$ is a global thresholding. The test is dependent only on the grey-level of the point.
(b) $T = T(N(x, y), f(x, y))$ is a local thresholding. The test is dependent on the neighbourhood property of the point and the grey-level of the point.
(c) $T = T(x, y, N(x, y), f(x, y))$ is a dynamic thresholding. The test is dependent on the coordinates, the neighbourhood property and the grey-level of the point.

It is clear that the simplest approach of the three thresholdings is the global thresholding. It is only based on the global thresholding value and the grey-level of a test point, irrelevant to the position and any local property of the image.
Selection of an appropriate thresholding value $T$ is an important issue. Most commonly used techniques are based on the analysis of the grey-level histogram. If an image is well suited to binarisation, it will feature two or more very clear peaks which are separated by well defined troughs in its histogram. Figure 2.11 is an ideal grey-level histogram, called bimodal histogram generated by a high contrast image. In this case, the lowest point of the trough can be selected as the threshold.

![Figure 2.11 An ideal grey-level histogram [46]](image)

The dual or interval threshold operation is an approach by which a binary output image is produced by converting all grey-level values between two thresholding values $T_1$ and $T_2$ into 1, and all grey-level values outside this interval into 0. More thresholding approaches may be referred to [46 - 49].

2.3.1.2 Image Smoothing

The process of image smoothing [46] [49] is primarily used for diminishing spurious effects in an image which are caused by a poor sampling system or transmission channel. The image noises usually have high spatial frequencies and may be removed in the smoothing process. So low pass filters are often applied to the input image in order to allow the low spatial frequencies of the desired image to pass through while attenuating
the high spatial frequencies of the noise elements. The filtering process can be implemented by a convolution operation with a simple 3 x 3 or 5 x 5 window or mark.

For a continuous signal the convolution operation is defined as

\[
g(t) = f * h = \int_{-\infty}^{\infty} f(\tau)h(t-\tau)d\tau
\]

(2.5)

Obviously the output \( g \) of a shift-invariant linear system, such as most optical electronic systems and most filtering techniques, is given by the ‘convolution’ of the input signal \( f \) with a function \( h \) which completely characterises the system response known as the ‘impulse response’. The function \( h \) is normally referred to as a filter since it represents what elements of the input image (signal) are allowed to pass through to the output image (signal). By choosing an appropriate filter, one can enhance some parts of the output in a system and attenuate others. The two dimensional convolution operation is given as

\[
g(x,y) = f(x,y) * h(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta)h(x-\alpha, y-\beta) \ d\alpha \ d\beta
\]

(2.6)

Its discrete form or digital convolution can be written as

\[
g(x,y) = f(x,y) * h(x,y) = \sum_{i} \sum_{j} f(i,j).h(x-i, y-j)
\]

(2.7)

where, integer values \( i \) and \( j \) have replaced the linear variables \( \alpha \) and \( \beta \). The filter function \( h \) with the form of two-dimensionally sampled points represents the continuous impulse response of the filter. The summation is taken only over the area where the image \( f \) and the filter \( h \) overlap. Thus the \( h(x, y) \) can be regarded as a window (mark,
kernel or template) which is moved over the complete digital image $f(x, y)$. A $3 \times 3$ convolution can be illustrated as in Figure 2.12.

![3x3 Convolution Filter](image)

**Figure 2.12** $3 \times 3$ convolution filter $h$ [47]

Neighbourhood averaging is one of the simplest approaches to image smoothing. This filter is a $3 \times 3$ window and its smoothing process may be expressed as

$$\text{Smoothed value} = \frac{1}{9} \sum (\text{Neighbourhood values})$$ (2.8)

It operates by replacing each pixel value with the average or mean of its immediate neighbours. This averaging process reduces the image noise but brings about the side-effect of smearing the significant grey-level differences (the edges) between pixels in a neighbourhood. The edge blurring may be partially overcome by setting up a threshold $T$ incorporated with the filter [46].

Gaussian smoothing is also one of the most commonly used smoothing techniques. Based on a Gaussian function, an image can be convolved with certain degree of
smoothing which is achieved by changing the convolution mask coefficients. The two-dimensional Gaussian function $G(x, y)$ is defined as

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left\{-(x^2 + y^2) / 2\sigma^2\right\}$$  \hspace{1cm} (2.9)$$

where $\sigma$ is the standard deviation for this distribution. The degree of smoothing is controlled by the shape of the Gaussian curve used, that is, it depends on the values of the standard deviation $\sigma$, see Figure 2.13.

![Figure 2.13 Gaussian curves with different $\sigma$ [46]](image)

In application, two-dimensional Gaussian filtering can be achieved by convolving the image with a one-dimensional Gaussian mask in the vertical direction first and then the horizontal direction. Generally the sizes of the marks are vary large in Gaussian filtering and so its processing speed is often too slow for on-line applications. Other smoothing filters may be referred to [46] [49].

### 2.3.1.3 Edge Detection

An edge is a boundary between two regions which have different grey-level values. Edge detection is by far the most common approach for detecting meaningful discontinuities in
grey-level in the process of image segmentation. There are great number of techniques available for edge detection. Here only those operators based on the gradient are reviewed.

The gradient of an image \( f(x, y) \) at point \((x, y)\) is a two-dimensional vector and given by

\[
\begin{bmatrix}
  G_x \\
  G_y
\end{bmatrix} =
\begin{bmatrix}
  \frac{\partial f}{\partial x} \\
  \frac{\partial f}{\partial y}
\end{bmatrix}
\]

(2.10)

which represents the rates of change of the image \( f(x, y) \) in the \( x \)- and \( y \)-directions respectively. The magnitude of this vector which is involved in the edge detection, is given as

\[
G[f(x, y)] = \left( G_x^2 + G_y^2 \right)^{1/2}
\]

(2.11)

This quantity is equal to the maximum rate of increase of \( f(x, y) \) per unit distance in the direction of \( G \). The gradient can be approximated by absolute values:

\[
G[f(x, y)] \approx |G_x| + |G_y|
\]

(2.12)

The direction of the gradient is given as

\[
\alpha(x, y) = \tan^{-1}\left( \frac{G_y}{G_x} \right)
\]

(2.13)

where the angle is measured with respect to the \( x \) axis.
In the discrete domain of digital images the partial derivatives above become simple first differences. For example, the first difference of \( f(x, y) \) can be expressed as, in the x-direction

\[
f(x + 1, y) - f(x, y)
\]  
(2.14)

in the y-direction

\[
f(x, y + 1) - f(x, y)
\]  
(2.15)

The Roberts gradient using cross-differences in a 2 x 2 region can be written as

\[
G[f(x, y)] = ([f(x, y) - f(x + 1, y + 1)]^2 + [f(x + 1, y) - f(x, y + 1)]^2)^{1/2}
\]  
(2.16)

The Roberts operator is sensitive to noise [47], so 3x3 region operators such as Sobel and Prewitt operators are utilised. Using the 3x3 region for computation of the gradient has advantage of increasing the quality of edge detection and reducing its sensitivity to noise. The Sobel operator estimates the partial derivative in the x-direction by

\[
G_x = [f(x + 1, y - 1) + 2f(x + 1, y) + f(x + 1, y + 1)] - [f(x - 1, y - 1) + 2f(x - 1, y) + f(x - 1, y + 1)]
\]  
(2.17)

Similarly in the y-direction:

\[
G_y = [f(x - 1, y + 1) + 2f(x, y + 1) + f(x + 1, y + 1)] - [f(x - 1, y - 1) + 2f(x, y - 1) + f(x + 1, y - 1)]
\]  
(2.18)

In an analogous manner Prewitt operator gives the partial derivatives by

\[
G_x = [f(x + 1, y - 1) + f(x + 1, y) + f(x + 1, y + 1)] - [f(x - 1, y - 1) + f(x - 1, y) + f(x - 1, y + 1)]
\]  
(2.19)
\[ G_y = \{ f(x - 1, y + 1) + f(x, y + 1) + f(x + 1, y + 1) \} \]
\[ - \{ f(x - 1, y - 1) + f(x, y - 1) + f(x + 1, y - 1) \} \]

It is common to estimate the directional differences using simple convolution masks, one mask for each different operator \( G_x \) or \( G_y \). Convolving these masks with an image produces the gradient \( G \) at all point in the image. The convolution masks for the Roberts, Sobel, and Prewitt operators are shown in Figure 2.14. Once the gradient magnitude has been estimated, a decision of if an edge exists is to be made through a predefined threshold. The threshold may be selected by a trade-off between valid and false edges.

Figure 2.14 Convolution masks: (a) Roberts; (b) Sobel; and (c) Prewitt edge detection operators [47]
2.3.2 Model Based Object Recognition

In application of robotic assembly, a vision or sensory system is utilised to identify and locate a specified object in the scene. In this case, the vision system must have full knowledge of the desired object. The *a priori* knowledge of the object is provided through a model of the object. Generally the model contains the geometrical knowledge of the object, and in some cases it may include the functional information of the object. A vision system which makes use of an object model is referred to as a model-based vision system. The process of identification of an object with respect to its object model is referred to as a model-based object recognition. A general paradigm in model-based computer vision system for object recognition is illustrated as in Figure 2.15.

![Figure 2.15 A general paradigm in model-based vision [50]](image)

The data collection and low level process are included in discussion of the scene constraints, image acquisition, pre-processing and segmentation in Figure 2.11. The data description concerns how to describe the shape of an object, and extract the object features from its image data in order to match those features with the information embedded in the object model. In the following, the useful object features and the methods for object modelling and recognition are briefly introduced. Approaches to
object representation, as aforementioned, include a CSG tree, B-Rep modelling and feature-based representation. More applicable representations for description of object shapes are referred to [50].

2.3.2.1 Feature Extraction

The feature extraction is used for identifying inherent features of an object in an image. These features describing the object or attributes of the object can uniquely identify key differences between objects in applications. Although there are wide range of methodologies available for selection of features, it can be generally noted that features should be completely invariant with respect to the translation (position), scale (size) and orientation (rotation) of the objects. Many of the features of interest are concerned with the shape of an object, such as area, perimeter, compactness (perimeter²/area), centre of gravity of the image, and so on. Figure 2.16 shows the parameters of area, perimeter and compactness for various shapes.

Figure 2.16 Features for various shapes [46]

The compactness is a useful measure of elongation of shapes. It may be expressed as

\[ C = \frac{P^2}{(4\pi A)} \]  

(2.21)
where \( P \) is the perimeter and \( A \) is the area of the region of the shape. For an ideal circle its value is equal to 1. If the value is greater than 1, it means the shape under measuring resembles less than a circle. So it is possible to use the compactness to roughly distinguish a circle from other shapes.

Moments are useful tools of describing the properties of an object in terms of its area, position, orientation and other precisely defined parameters. They are invariants under change of size, position, rotation and/or reflection. For the shape of an object in an image, the moment of the shape can be defined as [46]

\[
m_y = \sum x \sum y \cdot x^i \cdot y^j \cdot f_{xy}
\]  
(2.22)

where the order of the moment is \( i + j \). The \( x \) and \( y \) are the pixel coordinates and \( f_{xy} \) represents the pixel brightness. Zero- and first-order moments can be given as

\[
m_{00} = \sum x \sum y \cdot f_{xy} \quad m_{10} = \sum x \sum y \cdot x \cdot f_{xy} \quad m_{01} = \sum x \sum y \cdot y \cdot f_{xy}
\]  
(2.23)

In a binary image since \( f_{xy} \) is either 0 (black) or 1 (white), \( m_{00} \) represents the area of the shape. The centroid of the shape specifying the position of the object can also be expressed in terms of moments as

\[
x_c = \frac{m_{10}}{m_{00}} \quad \text{and} \quad y_c = \frac{m_{01}}{m_{00}}
\]  
(2.24)

With respect to the centroid, the central moment \( \mu \) can be calculated by

\[
\mu_y = \sum x \sum y \cdot (x - x_c)^i \cdot (y - y_c)^j \cdot f_{xy}
\]  
(2.25)
then the normalised central moments $\eta$ is given as

$$\eta_i = \frac{\mu_{ij}}{(\mu_{00})^\lambda} \quad (2.26)$$

where $\lambda = \frac{(i+j)}{2} + 1$, and $(i+j) \geq 2$. Obviously $\mu_{00} = m_{00}$. A set of central moments and the derived invariant moments in terms of the normalised central moments $\eta_i$ are referred to table 6.1 of [46]. More methodologies of feature description and extraction may be found in [46] [49].

2.3.2.2 Object Modelling and Recognition

Object models are generally involved with \textit{a priori} geometrical, topological and even functional knowledge of the objects under investigation. There are two main approaches to object modelling [50]: sensor-based and through a CAD/CAM system or a user-developed system similar to the description of CAD systems. In the sensor-based approach, multiple viewpoints of an object are required for constructing a complete geometrical model of the object. This approach is much sensitive to image data noises, and needs techniques developed for registering the data from all possible viewpoints and then incorporating the data into the model. Using CAD systems to construct object models has many advantages over the sensor-based approach. Object geometrical information can be described in detail in the object models, and object functional knowledge can also be included in the models with a CAD database or knowledge base. Furthermore, the object data in the models can be integrated with product design, manufacturing, production control and management through a CAD/CAM system or a computer-integrated manufacturing system. Examples of these CAD systems incorporated for object recognition may be found in [51-55].
The process of object recognition is to establish a correspondence between object models and an image data taken from a vision system, that is, to match the features extracted from the image data to those derived from the appropriate object models. There are a large number of methods for applications of object recognition. De Floriani [56] proposed a generalised edge-face graph (GEFG) to describe the object boundary and identify the object features. Marefat and Kashyap [57] use cavity graphs, which provide a geometrical and topological description of the depressions of objects, for recognition of 3D object features. Based on CAD models, Zhang et al [58] described a method to automatically construct a relational model using a hypergraph for 3D object recognition. During the recognition phase, a best-first search is applied in the graph for acceptable match. Flynn and Jian [55] also use the relational graph models to match object features in terms of CAD models. Wang and Iyenger [59], and Fan et al [60] take surfaces as object features derived from object models for recognition purposes. A heuristic search is applied to find the matching of the object surfaces in Wang and Iyenger's approach. In Fan et al's method the matching is performed through three modules: the screener, the graph matcher, and the analyzer.

Many applications of moments are taken for recognition of objects. Dudani et al [61] used the moment invariants for identifying different shapes of aircrafts, Ei-Khaly and Sid-Ahmed [62] for character recognition, and Wen and Lozzi [63] for recognition and inspection of manufactured parts. Luo et al [64] [65] applied the moment invariants to robotic application. In their system the shape features are extracted from the combination of tactile and visual information based on the moment invariants.

2.4 Group Technology and Its Application

Group technology (GT) as a manufacturing concept, is widely applied in many areas of industry. In this section, the principle of the group technology, its major schemes for parts classification and coding, and its applications are briefly reviewed.
2.4.1 Principle of Group Technology

Group technology is a manufacturing philosophy which can be applied to increase production efficiency, reduce production cost by sharing common processing equipment and using common operations. Using GT similar parts are grouped together and classified into a same family. Parts classification may be made in terms of their similar design or manufacturing characteristics. Figure 2.17 is shown the thirteen parts with similar manufacturing process requirements but different design characteristics.

![Figure 2.17 Parts similarity in manufacturing [3]](image_url)

In order to group parts into families, parts classification and coding is required which is concerned with identifying the similarities among parts and relating these similarities to a coding system. The coding system represents the part families with similarities encoded in the system. A part family collects parts which are similar either in their geometrical
shape and size or in their similar processing steps required in manufacturing. The parts within a family are different, but their similarities are close enough to distinguish them from those parts in other families in terms of their characteristics identified and classified. The parts shown in Figure 2.18 might constitute a part family in manufacturing, but may not be grouped as a design part family in terms of their geometrical characteristics.

2.4.2 Classification Methods

There are three methods of grouping parts into families. They are [3]

1. Visual inspection
2. Production flow analysis (PFA)
3. Parts classification and coding system

Using the visual inspection method, one can, according to his knowledge and experience, classify parts into families by looking at the physical parts or their photographs. Obviously this method is less accurate and much dependent on user's professional knowledge. The second method, PFA, groups parts together in terms of their similar operation sequences and machine routings through analysing the route sheets of parts. This method is not used much in industry. The third method, parts classification and coding, is generally considered to be the most powerful method of the three and hence is widely used in design and manufacturing. In this method of grouping parts into families, the design and/or manufacturing attributes of each part are examined and identified by means of a code number or symbol. A parts coding system consists of a sequence of symbols that identify part's attributes. The symbols in the code can be all numeric, all alphabetic, or a combination of both types. The classification and coding systems have three basic code structures used in applications.
1. Chain-type structure
2. Hierarchical structure
3. Hybrid structure, a combination of hierarchical and chain-type structures

In the chain-type structure, as shown in Figure 2.18, each attribute is represented by each digit in each position. The representation of each digit is fixed in the sequence of the code and does not depend on the value of preceding digits. Therefore, this code, also called polycode, can become quite lengthy. The polycode provides relatively easy search capability since data can be extracted for selected position from its coding sequence. This can be helpful in recognizing specific part attributes for further applications.

![Figure 2.18 Chain-type code](image)

In the hierarchical structure, as shown in Figure 2.19, the meaning of each digit is dependent on the meaning of its preceding digits. The structure, commonly named monocode or tree structure, provides a relatively compact structure in which a large amount of data can be presented by a few digits.
In order to take advantage of both monocode and polycode, the hybrid structure is presented as shown in Figure 2.20. The hybrid code is typically composed of a series of short polycodes. Within these short polycode chains, the monocode is used to classify parts into groups.

The best known classification and coding system is the Opitz coding system developed by H. Opitz at the Technical University of Aachen, Germany. The system, as shown in
Figure 2.21, consists of five main digits with four supplementary digits. The first five digits represent the shape, i.e. geometrical attributes of a part and the four supplementary digits are used to indicate some particular attributes for manufacturing. The Opitz code can be extended by four additional digits called secondary code which may be used for identifying the production operation type and sequence. The details of the Opitz code may be referred to [3] [66].

<table>
<thead>
<tr>
<th>Digit 1</th>
<th>Digit 2</th>
<th>Digit 3</th>
<th>Digit 4</th>
<th>Digit 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part class</td>
<td>Main shape</td>
<td>Rotational machining</td>
<td>Plane surface machining</td>
<td>Additional holes and forming</td>
</tr>
<tr>
<td>L/D ≤ 0.5</td>
<td>External shape element</td>
<td>Machining of plane surfaces</td>
<td>Other holes and teeth</td>
<td></td>
</tr>
<tr>
<td>L/D &gt; 0.5</td>
<td>Internal shape element</td>
<td>Rotational machining</td>
<td>Other holes and forming</td>
<td></td>
</tr>
<tr>
<td>L/D &gt; 2</td>
<td>Main shape</td>
<td>Main bore and rotational machining</td>
<td>Machining of plane surfaces</td>
<td>Other holes and forming</td>
</tr>
<tr>
<td>Special</td>
<td>Main shape</td>
<td>Machining of plane surfaces</td>
<td>Other holes and forming</td>
<td></td>
</tr>
<tr>
<td>A/B &lt; 3</td>
<td>Special</td>
<td>Special</td>
<td>Special</td>
<td></td>
</tr>
<tr>
<td>A/B &gt; 3</td>
<td>A/C &gt; 4</td>
<td>A/B &lt; 3</td>
<td>A/C &lt; 4</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.21 Basic structure of the Opitz coding system [3]

The clustering approaches to grouping parts into families may be referred to Kusiak et al and King’s work [67 - 71]. These approaches are mainly concerned with determination of clusters of part families and machine cells based on production requirements.
2.5 Robotic Grippers

A gripper, hand or end-effector, is the device mounted at the end of a robot arm. With the gripper, workpieces or objects can be picked up, held, manipulated, transferred, placed and released accurately in a discrete position. In order to perform grasping tasks properly, it is required to take into account the grasp modelling and planning, the geometrical analysis of objects to be grasped, the gripping force and control of grasping as well as design of grippers etc. These issues are briefly reviewed in the section.

2.5.1 Grasp Modelling and Planning

To choose an appropriate grasping model is an important issue in determination of grasp and manipulation. Cutkosky et al [72-74] studied grasping and manipulation in terms of human grasping. They constructed a grasp taxonomy based on observations of single-handed operations by machinists working with metal parts and hand tools. These grasp models are influenced by grasping tasks (forces and motions) and object geometry. It is found that both task requirements and object shapes play important roles in choosing a grasp. They also studied analytic approaches to grasp modelling and grasp choice. The comparison of human and analytic grasp was made in their studies. Iberall and MacKenzie [75] [76] also studied human behaviour and addressed the question of how best to use hand features in relation to the anticipated object properties and predicted interaction outcome, in order to achieve task goals. Using the opposition space model discussed in their papers one can analyse human hand functionality in terms of task requirements. This provides a way to design and control versatile dextrous robot hands. Bekey et al [77] investigated six generic grasp modes and developed a knowledge-based system for selecting the appropriate grasp for spherical and cylindrical objects for different tasks.

Grasp planning is concerned with task requirements, object geometry and grasping manipulation as well as grasping environment. In order to realise an automatic grasp
planning, Dubey and Nnaji [78] proposed a knowledge-based grasp planner to reason out the optimal gripper type, grasp type and grasp characteristics from the assembly tasks, object features and its geometry. Feature reasoning is made to find an optimal grasp. The grasp is evaluated by the factors of manipulability, torquability, rotatability and stability. Laugier [79] also developed an approach to the automatic determination of grasping data. The approach is based on geometrical reasoning at successive levels of abstraction guided by heuristics which are described as filters. If a geometrical pair (face-face or face-edge) of an object passes the filters, it is taken as the grasping position set. Other relevant grasping data can also be obtained through the geometrical reasoning. Troccaz [80] studied computational geometry for grasping reasoning. In this reasoning process, a grasp may be related to two different levels of representations: a symbolic grasp and a numeric grasp. The symbolic grasp is object-oriented which describes geometrical features of an object when it contacts with a gripper. The numeric grasp is gripper-oriented which exactly describes the hand configuration of the gripper. Based on geometrical knowledge of objects and their environment, Wolter et al [81] addressed the problem of how to determine gripping positions in order to constitute a good grip. Some criteria in this regard were discussed in their paper. The grasp planning for multifingered robot hands may be referred to Li and Sastry [82] [83]. They [82] proposed three quality measures for evaluating a grasp. An optimal grasp may be obtained by analysis in terms of these three quality measures. In [83] they studied grasp planning and coordinated manipulation by a multifingered robot hand under various contact constraints. Control schemes for the multifingered hand were derived.

2.5.2 Sensor-Based Grasping Analysis

Robot actions such as grasping may be performed intelligently with sensors implemented with the robot gripper. Benhadj et al [84] developed a ‘Hand Eye’ system integrating a two-dimensional vision system with a pneumatic proximity-to-tactile sensing device. The system can recognise objects on-line and provide object data for robot handling based on an expert system. Mehdian and Rahnejat [85] presented an algorithm to determine the
orientation of various objects lying randomly regardless of their identity and find the stable hold sides of an object using tactile data from a sensory gripper. The optimal grip force may also be determined through the analysis of the tactile data. The review of the state of the art of tactile sensors may be referred to Foroughi et al [86].

A multi-proximity sensory system for the guidance of robot end effectors were described by Andre [87]. The sensory system is composed of infrared, ultrasonic and magnetic sensors. An interactive simulation software has been developed for computer aided design of sensors, sensitive end effectors, and relevant control algorithm. A sensor controlled gripper described by Dillmann [88] is a multi-sensory device with a combination of tactile and non-tactile sensors equipped. The tactile sensors work as force sensing wrist, grasp force sensor and touch sensors, and the non-tactile ones are applied for sensing the approach of objects and recognition of the object positions between fingers.

In order to achieve a given manipulation task of gripping in a partially structured environment, a combination of partial geometrical models and of vision data is used in the system developed by Laugier et al [89]. Three processing phases aimed at selecting a viewpoint avoiding occlusion, modelling the local environment of the object to be grasped, and determining the grasping parameters are taken in the system. Tzafestas [90] illustrated an integrated sensor-based intelligent robot system. In the process of gripping, a vision system provides an initial estimate of object position and orientation. And then tactile sensors confirm, modify or improve the initial information provided by the vision system.

2.5.3 Grasping Force Analysis

A stable grasp is concerned with the position of an object to be grasped and the force to be imposed on the object. The material properties of the object also influence the force determination. For a gripper the general requirements of the grasping force are that the
gripper should produce enough force to overcome the weight of the part and to keep the part in its grasp during the handling process. On the other hand, the gripper should not produce excessive force which may damage the object in some situations.

Chen [91], through the studies of many types of mechanical grippers, established the input-output force relationship of the grippers and determined the safe grasping force as the posture between the gripper and the object is changed. The relationship and the grasping forces can be obtained by the classical method of static force analysis. Based on mobility and connectivity analysis, Salisbury and Roth [92] investigated the kinematic structure and the gripping force of robot hands. This leads to a design of the Stanford/JPL hand. Discussion of grasping and manipulation forces for multi-fingered robot hands is referred to Yoshikawa and Nagai [93].

2.5.4 Gripper Structure and Design

There are various ways of classifying grippers and their actuating mechanisms. From their functions and finger types, grippers may be classified into two, multiple finger, outside and inside grippers, vacuum and magnetic devices, and dextrous hands which closely resemble the human hand in appearance and in the ability to touch and grasp. Here only the types of mechanical mechanism grippers are reviewed.

Commonly used two- and three-fingered grippers are shown in Figure 2.22.
In Figure 2.22 (a) and (b) are two-fingered grippers where (a) is designed to grasp an object on its outer surface and (b) used for both out and inner surface grasping. (c) is a two-fingered angular gripper with protective fingers. (d) is a three-fingered gripper suitable for gripping of round and spherical parts.
Two types of dextrous robot hands are shown in Figure 2.23.

Figure 2.24 (a) is the Utah/MIT dextrous hand with three fingers and a thumb designed by the University of Utah and the Artificial Intelligence Laboratory at MIT. The robot hand is used as a general purpose research tool for the study of machine dexterity. Figure 2.24 (b) is the Belgrade/USC dextrous hand with five fingers (including a thumb) developed by cooperation of the University of Belgrade and the University of Southern California, based on the Belgrade prosthetic hand. This hand is an anthropomorphic end effector for robot manipulators for grasping tasks.

In order to perform handling tasks, a gripper must be designed to suit the shape, mass and other physical and functional demands of the workpiece being handled, as well as the parameters of the handling and of the environment. The general requirements of a gripper design may be referred to Schneider and Servis [96]. Chen [97] provided an overview of gripping mechanisms for industrial robots. These gripping mechanisms,
according to kinematic pairs, can be classified into linkages, gears, cams, screws and flexible bands. In addition to the ordinary grippers, several types of versatile grippers such as soft fingers and anthropomorphic three finger gripper were discussed in his paper. Many kinds of grippers and their design and control may be found in [98-103]. The structure design and control of those dextrous robot hands shown in Figure 2.24 may be referred to Jacobsen et al [94] [104] and Bekey et al [95].

2.6 Conclusion

In this chapter, a number of issues of automatic assembly, image processing and object recognition, group technology and gripper design and grasping planning related to our project have been reviewed.

Representation plays an important role in grasping design and assembly analysis. In the three conventional representations of CSG, B-Rep and feature-based description, the feature-based approach is more flexible and easily implemented in applications since the features can be defined in different ways to suit different application purposes. Object models may be described in terms of feature representation. Knowledge-based representation is another way to represent design, assembly or grasping systems. Domain knowledge in grasping and system performance may be represented and inferred based on knowledge representation. In our system developed, the feature- and knowledge-based descriptions are integrated with a coding system in terms of GT technology. GT provides a means to identify and classify objects and processes in design and manufacturing. Integrated with the feature and knowledge representations, GT becomes more effective and significant for classification of grasping families.

There are many methods in image processing and object recognition. Using binary image data makes the processing much simple. Some basic processing approaches of thresholding, image smoothing, edge detection are discussed in the above review. In order to get useful information from an object image, moment invariants are taken as
object shape features in the process of feature extraction. These shape features are prestored in the system as templates (models) to match those derived from the image. On matching successful, the object is recognised. This is so called the model-based object recognition.

Grasping planning and operation are mainly concerned with the grasping tasks and the object geometry to be grasped. To find a proper grasp is a major concern in most investigations reviewed above. One should deals with the geometrical analysis of objects to build gripping models. This may be done by geometrical reasoning or knowledge inference of gripping or incorporation of both. Gripper design should meet object geometry, force and motion requirements. It is better to use a generic gripper to perform more grasping tasks.
CHAPTER 3  MODEL BASED OBJECT RECOGNITION

3.1 Introduction

Model-based object recognition is a strategy of recognizing and classifying objects based on their models which collect the objects information from visual sensors or CAD data or from both of them. The object models may be represented by aforementioned methods of B-Rep, CSG or feature-based representation. Object features may be chosen with respect to the object shape, surfaces, local properties [105], volumetric structure, or functional properties. To design such a model-based object recognition system, it is required to determine the following issues: (1) the type of sensors for data collection, (2) selection and extraction of object features for recognition, (3) the approach to representation of object features for constructing object models, and (4) the method of matching of sensory data to that of models.

In this study, a recognition system is designed for the purposes of extracting object gripping features and performing the grasping tasks of assembly. The recognition system uses infrared sensors to collect two-dimensional object data in the scene, and integrates the sensory data and two types of models, namely the object model and the gripping model constructed in the system, to extract the object gripping features. The gripping model is coupled with a coding system which is proposed based on group technology for object classification for automated assembly. The coding system describes object features relative to grasping operations and it is incorporated with the gripping model to analyse and classify objects into specific gripping families. A tree search algorithm [50] is applied to the matching procedure and the matching search process is carried out in terms of a rule-based knowledge representation.

In the next section, the recognition system is outlined and described briefly. Section 3.3 provides a detailed discussion of image data collection and processing. Section 3.4 discusses selection and extraction of object features as well as the approach and process
of object recognition. The experiment setup is given in Section 3.5, and finally summary and conclusion are made in Section 3.6. The proposed coding system is discussed in next Chapter.

3.2 Outline of the Recognition System

The schematic of the model-based object recognition system is shown in Figure 3.1.

![Figure 3.1 System schematic of model-based object recognition](image)

When an object is presented in the scene, its image (sensory) data are collected by a visual system which is made up of infrared sensors. Since analysis of a 3D sensory data is much more difficult and time-consuming, the surface-based representation of 2D shapes of an object is applied. We use surfaces as the object geometric features for recognition. The different shapes of the object surfaces are identified in the feature extraction process. Then the features from the scene are matched with those stored as models. The successful match is interpreted into a digital code through a knowledge-based inferencing
algorithm. At the same time, a suitable gripping model is set up for the object to be grasped.

3.3 Sensory Data Collection and Processing

As mentioned above, the sensory data of an object in the scene are collected by means of a sensor made up of infrared photodiodes. Then, in order to diminish the sensory noise in the image data, some necessary low-level processings, such as thresholding, smoothing, are utilised. Furthermore, edges in the data are detected for feature extraction and object recognition.

3.3.1 Sensor Structure and Data Collection

The sensor in the system is designed by twenty five infrared sensors in which nine of them are infrared emitters and the other sixteen are infrared detectors (photodiodes). The action of photodiodes are based on the photoelectric effect discovered by Hertz in 1887 [106] [107]. A photodiode is made as a PN or PIN junction using special semiconductor materials. The action principle of the photodiode is that, when the energy of light strikes the diode, hole-electron pairs are generated which produce an electric current that is proportional to the intensity of the light incident on the junction.

The sensor designed in the system is shown in Figure 3.2, where the infrared tubes are arranged in two straight lines: one line is for emitters and the other one for detectors.
When the sensor moves over an object, the object image data are collected and formed an array. Figure 3.3 shows the data array of a block obtained by scanning the block using the sensor. The sensory array of discrete points is two-dimensional data collected from the top surface of the block, where the higher values indicate the object and lower values the background. In order to get better data, it is required that the scanning plane of the sensor is parallel to the working plate where the object is put on.
The range of the image data value is from 0 to 10 set by an Analogue-to-Digital Converter (ADC). The emitters of the sensor project lights (infrared rays) on to the object in the scene, and at the same time the detectors receive the reflection lights both from the object and its background. The higher values in the image are collected from the object, while the lower value from its background.

### 3.3.2 Data Processing

Data processing is to diminish the noise produced in the sensing process and enhance certain desirable data of interest from the image based on grasping tasks. The sensory data above may be expressed as a two-dimensional discrete function \( f(x, y) \). In the image data processing, the techniques of thresholding, smoothing and edge detection are employed in \( f(x, y) \).
3.3.2.1 Thresholding

In our system, a global threshold is applied and it is used mainly for generation of a binary image. In our application, the binary image provides information enough to represent the object shape two-dimensionally and also much simplifies the image processing. The thresholding value is determined by analysis of the histogram of the object image data. The histogram can, at this stage, give some useful information about the nature of the image. Figure 3.4 illustrates three histograms of three different image data taken from a same object - the block mentioned above. The first histogram comes from the data shown in Figure 3.3.

It is clear that each histogram can approximately indicate the degree of contrast within the image in a different possible levels. From the observation of these histograms, one may choose a value between 5 and 8 as the threshold to separate the image. After using a try and error process, the value 6.5 is chosen as the threshold to transform an original image \( f(x, y) \) to a binary image \( g(x, y) \). This can be expressed as

\[
\begin{align*}
\text{IF} \quad f(x, y) \geq 6.5 & \quad \text{THEN} \quad g(x, y) = 1 \\
\text{IF} \quad f(x, y) < 6.5 & \quad \text{THEN} \quad g(x, y) = 0
\end{align*}
\]  

(3.1)

By using the above binarizing operation to each point of the image in Figure 3.3, the binary data are obtained as shown in Figure 3.5.
Figure 3.4 Image histograms of a block
The processing of thresholding operations above is made by mapping the value of each point (pixel) in the original image directly and uniquely to a new value in the transformed image. However, used smoothing algorithms, the transformed value is regarded as a function of a group pixel values in some specified spatial location in the original image. It is convenient to use a local area centred on the pixel to be calculated and to achieve the transformation. The neighbourhood averaging defines the local area as a window (mask) to perform the data smoothing by convolving the original image with the mask. A 3x3 neighbourhood average mask is shown in Figure 3.6 which is applied in our system.
Figure 3.6 3x3 neighbourhood average mask

Fairhurst [48] indicates that a smoothing function in conjunction with a thresholding operation, when it operates on a binary image, can be quite effective in removing spurious noise points and in cleaning up the image in order to produce the transformed image in which the shape or geometrical features of the object may be well defined. Such a smoothing function may be specified as

\[
g'(x,y) = \begin{cases} 
1 & \text{if } \frac{1}{9} \sum (\text{Neighbourhood values}) \geq T \\
0 & \text{if } \frac{1}{9} \sum (\text{Neighbourhood values}) < T
\end{cases}
\] (3.2)

where T is some chosen threshold. In our cases, the T is chosen as 4.4. By using the above smoothing function, the binary image shown in Figure 3.5 is transformed into that as shown in Figure 3.7.
3.3.2.3 Edge Detection

Boundary edges can provide useful information on the shape and geometry of an object. The aim of edge detection is to search for discontinuities in terms of image intensity by examining the local neighbourhood distribution of pixel values. In practice, a window is used as an edge detection operator, as discussed in Section 2.3, which is moved across successive image points, at each point by computing the value based on local neighbourhood information to find the most appropriate place in which to locate the edge between object and background.
Using a gradient function to identify the intensity discontinuities is a common method in edge detection. The Sobel operator on the basis of the gradient in Figure 2.15 (b) is regarded as one of the best simple edge operators. Its advantage is that, in addition to its primary function in evaluating a gradient value in the local area specified by the window, it contains a degree of inherent neighbourhood smoothing. Used the Sobel operator, the image in Figure 3.7 is detected and its boundary edges are identified as follows (Figure 3.8).

![Image data after edge detection](image)

**Figure 3.8** Image data by edge detection

### 3.4 Object Features and Recognition

Features have been recognised as a natural form of representation of objects. They are usually catalogued into form (or shape) features and functional features, where the form features capture the geometrical and topological properties of objects and the functional
features specify functions when these objects are involved in design, handling, manufacturing, assembly and so on. In our applications, features are defined in terms of two application oriented. One is for recognising objects in the scene and the other for capturing grasping features of objects of interest, with respect to object models and gripping models respectively. In this section, only those features relative to recognition are described.

3.4.1 Object Models and Shape Features

An object model is used to describe its geometric characteristics. A face feature-based representation is developed for construction of the object model. A face is a part of the surface bounded by edges of an object. Faces are regarded as the principal features in the model construction and grasping analysis. And they may be classified according to their geometric and grasping characteristics in Figure 3.9. The Primary face of the object is selected as the grasped face that has a contact area with the tip of the gripper required for the grasp performance. The secondary face, however, is less important in the grasping operation, but it may have same classification as the primary face.

![Face classification](image)

In previous work, such as Chen et al [7] and Nitschke et al [10], the object model is constructed or reconstructed by features through a feature graph or feature relation.
graph which specifies the feature relations as Is_In, Is_On, Adjacent_To, Dependent_Upon, and so on. In their models, the object geometry connected with features is represented completely. However, in our application of grasping design, it is not necessary to describe the object geometry in detail. In this system, the concept of a distinguished shape is proposed and complemented for object modelling, in addition to for object recognition.

Figure 3.10 presents the shapes of two different parts viewed from top and sides directions. For object-1, face 2 and face 3 may be chosen as the gripping elements, since they are regarded as the primary features of the object. For object-2, its primary features may be face 3 and face 4. In many cases, objects can be distinguished through their top views. In our system, we define the distinguished shape (or surface) as that it can identify the object in the scene. The difference between the distinguished shape and the primary feature (surface) of an object is, that the former is used to distinguish this object from others, while the later is related to the grasping performance. In certain situation, the distinguished surface could be taken as the gripping element.
An object model is constructed in such a way that the distinguished face is regarded as the centre of the object modelling and then other faces adjacent to it and their (gripping) relations are taken into account, on assumption that the object is properly fed into the scene for gripping and the top view of the object represents its distinguished shape in which there are no grasping elements existing (Side views could be presented if the distinguished shape matching could not be made by the top view in certain situations.). Based on this modelling strategy, the structure of the object model is much simplified and the gripping information is easily embedded in the model. The model is internally represented in three hierarchical levels shown in Figure 3.11. The top level is the face relation graph similar to that feature relation graph illustrated by Nitschke at al [10], but there is difference between them in that the face relation graph consists of faces and their relations which are potential grasping elements, while the feature relation graph is constructed by all features of an object. The second level contains the properties of each face in the graph. And the third level gives the detailed descriptions of face geometry.
In the face relation graph, nodes in the graph indicate the distinguished top face and its adjacent faces and arcs indicate the relations between the faces. The relations of the faces may be parallel or cylindrical, etc. Figure 3.12 depicts two face relation graphs of the objects shown in Figure 3.10. For the object-1, face 1 is adjacent to two pairs of faces which are parallel; for the object-2, face 1 that is part of the distinguished shape from its top view, is adjacent to three pairs of parallel faces in which the face 4 and face 5 constitute a step. The face relation graph proposed in our system has the following characteristics:

**Figure 3.11** Object modeling with three levels
1. It is constructed in terms of the face features of an object;
2. It centres around the distinguished face of the object and the distinguished face has only adjacent relation with other faces;
3. The adjacent relations among other faces are excluded in the graph;
4. The bottom part and other features like the step in Figure 3.12 are not taken into account.

The face relation graph acts as a bridge connecting the image data on its one side and the gripping features on the other side. In the face properties level, mainly the faces types are defined such as planar, cylindrical, etc. Whether a pair of faces are primary faces or secondary ones or not is dependent on if the pair is chosen as gripping elements. This is determined by considering the grasping and assembly characteristics of the object, which are represented as attachment to its object model. The grasping characteristics impose some constraints on selection of gripping elements, for example the distance between the two faces of a pair, while the assembly characteristics are mainly concerned with the
mating faces of the object in assembly. The third level contains the description of the geometrical details of the faces involved in the face relation graph. Only the dimensions of faces are described in the level. That is to say that each model is parameterized, where the parameters define the dimensions of the model. The values of the parameters of a model are extracted from the object data in the recognition process. When the correspondence of the principal features between the sensory data and the model is established, the actual values of the parameters are attached to the model and the model is then completely constructed.

As aforementioned, the distinguished shape is defined mainly for recognizing objects in the scene. The shape features of perimeter, area, compactness and eccentricity as well as its position and orientation are useful information for description and recognition of an object. These features may be represented by the moment invariants discussed in Section 2.3.2. In terms of a binary image, the shape area $A$ may be determined by the zero-order moment $m_{00}$ which equals the number of all pixels equal to 1; the perimeter $P$ may be obtained by simply counting all pixel points in the boundary edge; the compactness $C$ is calculated by $C = P^2 / (4\pi A)$ as mentioned before; and the eccentricity may be expressed by the zero-order and second-order central moments $\mu_{00}, \mu_{11}, \mu_{02}, \mu_{20}$ as [46]

$$E = \frac{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}{\mu_{00}}$$  \hspace{1cm} (3.3)

The object position may be given by the moments as $x_c = m_{10} / m_{00}, y_c = m_{01} / m_{00}$ as mentioned before, and its orientation may also be expressed in terms of the second-order moments as

$$\theta = \frac{1}{2} \tan^{-1} \left[ \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right]$$  \hspace{1cm} (3.4)
where $\theta$ is the orientation with respect to the x-axis of the object shape.

With respect to those features above, an integration of recognition searching and object modelling is made in a hierarchical structure similar to the searching tree introduced by Arman and Aggarwal [50]. Object models built in our system, as mentioned before, is not only for recognition purpose but also for providing data for establishment of object gripping models. The searching strategy and process based on the distinguished shape is detailed in next section.

### 3.4.2 Searching for Object Recognition

The searching strategy is that the search is performed within the object distinguished shape space and along a hierarchical searching structure designed for recognition, with combination of an object image data from the scene and its model information prestored in the system. Generally, the searching for object recognition may encompass two procedures. One procedure is to compute the moment invariants and extract shape features with respect to the digitized data of an image. The image data used in the system come either from the top view or the side view of an object. Figure 3.2 is an example of the digitized image data directly taken from the top view of the object-1 shown in Figure 3.9. The other procedure is to match the shape features extracted from the image with those features embedded in object models. The Euclidean distance measurement is employed to measure the similarity between the image and each feature vector set of models. The feature vector set $FV$ may be expressed as

$$FV = \{A_o, P_o, C_o, E_o\} \quad (3.5)$$

where $A_o$ stands for the area of the distinguished shape of the models; $P_o$ for the perimeter; $C_o$ for the compactness; and $E_o$ for the eccentricity. The feature vector sets of the models are generated off-line, in the process of model description in Figure 3.1.
Their data are obtained from the same view direction as the image. Assume that an image shape features are $A_i, P_i, C_i$, and $E_i$, then their Euclidean distance can be given by

$$ED = \sqrt{(A_i - A_o)^2 + (P_i - P_o)^2 + (C_i - C_o)^2 + (E_i - E_o)^2}$$  \hspace{1cm} (3.6)$$

The image shape features should be calculated with each feature vector set of models. Then the results of the Euclidean distance measurements are compared with each other, and accordingly the smallest value matches the image data and the desired object.

The searching is a knowledge based searching. The parameters representing the object shape features are embedded in the object model. The step of model description includes the searching knowledge which builds the relationship between the image features and model features. The searching starts from the top view of the object. If successful, the searching stops. Otherwise, it continues checking the side view of the object to find out its distinguished shape and achieve an appropriate matching.

The recognition searching is hierarchically performed as shown in Figure 3.13, with respect to the distinguished shapes of objects. The searching is mainly done through three steps: to extract the shape feature of the image, to distinguish circle and noncircle shapes, and to search the distinguished shape of the object. As soon as the distinguished shape is identified, the appropriate object is matched. The step of distinguishing circle and noncircle shapes is used to roughly classify objects into two different groups, prepare for the establishment of the gripping model and simplify the searching process.

The distinguished shapes are determined and described according to a priori knowledge of the known object. A single part may have more than one distinguished shapes from different sides of views. For example, for the object-2 in Figure 3.10 its right side view may also be selected as a distinguished shape. The determination principle of them is, any one of them can discriminate the object from others. An IF - THEN rule is used to
represent the relationship of the distinguished shapes with the object. As aforementioned, the top view is checked first. After the image data is processed, a searching is carried out in the distinguished shape space. Once the searching is successful, the object is recognized. Otherwise, a side view should be taken into account. The shape from side view is, at first, still searched in the distinguished shape space. If not successful, then the searching stops and the object is rejected. As mapping shapes into the distinguished shape space, the searching space and the searching time are much reduced eventually.

Figure 3.13 The recognition searching
After an object is successfully recognised, its geometrical data and its gripping model will be extracted from the object hierarchical model shown in Figure 3.11, by means of the face relation graph using IF-THEN inference.

3.5 Experiment Setup

The experiment setup for the object recognition is composed of an infrared sensor, a working table and a PC-30B board installed in a PC computer, refer to Figure 3.14. When an object is placed on the working table properly, its image data is obtained by scanning over it using the sensor. The analogue signal of the sensor is converted to the digitized data through the PC-30B board.

Figure 3.14 Experiment setup for object recognition
The infrared sensor is shown in Figure 3.2 and its structural prototype is given as follows (Figure 3.15).

The structure of the sensor consists of 1. emitters, 2. receivers, 3. a body, 4. supports and 5. an adjustable screw. The nine emitters and sixteen receivers are housed in the body which may be installed in a moving table by the two supports. The adjustable screw is used to adjust the emitters which can be moved backward and forward along their axes in order to get better lights projection.

The heart of the sensor is the emitters and receivers, where the beam of lights are transmitted by the emitters called light-emitting diodes (LEDs) and the image of an object is formed and displaced by the receivers called photodiodes through reflection of the transmitting lights. The distance between two photodiodes is 8 mm which provides a spatial resolution along the photodiodes line (x direction) in the sensory plane, refer to
Figure 3.16. The spatial resolution in y direction is dependent on the moving speed of the sensor when it is scanning over the object. The grey-level resolution is in the range of 0 to 10 controlled by the PC board.

The electronic circuits of an emitter and a receiver are given in Figure 3.17.
Figure 3.17 Circuits of the emitter and the receiver
When the transistor is selected, the light strength may be adjusted by changing the values of the resistances R1, R2 and R3. The output voltage Vo may be adjusted by changing the value of R4( = R5) or the ratio of R7 and R6.

The PC-30B board is a full size, low cost, high accuracy analogue and digital I/O board for the IBM PC and compatible series of computers. It features 16 analogue input channels with 30 KHz throughput, two 12 bit D/A outputs, two 8 bit D/A outputs and 24 digital I/O lines [108]. For the object recognition in our system, the 16 A/D channels are used for sending the output voltages of the 16 photodiodes into the computer. The major component of the A/D subsystem is a monolithic analogue to digital converter which accepts analogue voltage inputs from sensors and converts them into 12 bit digital codes. This process is carried out according to the software in use. In our system, Visual Basic is employed for the processing. The A/D system with 16 single-ended inputs can be configured for full scale input ranges either from 0 to +10 v (unipolar) or from -5 to +5 v and -10 to +10 v (bipolar). In our system, the range of 0 to 10 v is used correspondent to the output voltage of the photodiode. There are two operation modes: single conversion and continuous conversion in the A/D system. The single conversion is performed on the selected input channel and stopped on completion of this conversion. The continuous conversions are performed at a set rate. This rate is set by programming the board’s internal timer or an external clock source. In our system the continuous conversion mode is applied at an internal timer rate of 1.5 seconds.

3.6 Conclusion

In the chapter, the model-based object recognition system is described and the structure and function of the infrared sensor are illustrated. The recognition system has following characteristics.

1. The shape properties of an object are taken as image features for the object recognition.
2. The object models are described based on their faces and faces relations in three levels as shown in Figure 3.11.

3. The process of object recognition is a process of searching the distinguished shape features of the object.

In recognition, object shapes are regarded as features for object recognition and they are described by means of the moment invariants. The reason for this is twofold. Firstly, the shape image measured by the infrared sensor is in low resolution because of the small array size available. Therefore, such the shape data are not meaningful enough to be described by most boundary based shape descriptions such as Fourier descriptors, chain codes and so on. However, the moment invariants based on a shape region are regarded as an appropriate tool applicable for these kind of shapes. They are not required to describe the detailed features (local features, boundary details) of the image, but they are only employed to describe the properties of a whole shape and are accurate enough to identify the object in the scene. Secondly, for our robotic grasping applications, it is necessary to find out the position and orientation of an object to be grasped. The moment invariants turn out to be the best choice for this purpose.

The face relation graph is proposed to represent object models. The graph representation describes the faces and their relations which construct an object. But the graph does not take every face of the object into account. It only considers the potential grasping pairs (faces) to build the object model. This has advantage of easily extracting grasping features and obtaining a gripping model for the object to be grasped.

The searching process of object recognition is to find the distinguished shape of an object. The distinguished shape of the object predefined by the system tells its difference from other objects with respect to their geometrical characteristics. As aforementioned, limiting the searching in the distinguished shape space would increase the searching efficiency.
CHAPTER 4 OBJECT CODING SYSTEM

4.1 Introduction

For an object to be grasped, the information on its physical properties and geometrical configuration such as the weight, size, surface property and contact area as well as the position and orientation of the object, are required prior to the performance of the grasping operation. The information relating to the object may be obtained as a part of the object recognition process, refer to Figure 3.1.

There are two classical approaches for data extraction for grasping with a gripper. In the first approach, no prestored information about the object to be grasped is required, and the object data are obtained on line from a sensory system. In this case, a large number of computations should be carried out on-line. The object features for a suitable grasp are extracted based on the above data analysis. The second way is to build a model of the object beforehand. All the features of the object are described in detail within the model. A suitable grasp is determined only through the data analysis of the model. Geometrical computations are done off-line in this case. A combination of the two approaches has been studied and satisfactory results have been reported [50] [109].

In this study, the combination of the geometrical model with the sensory information approach is made in such a way that only principal features of the object which are related to the grasp, are utilised. The method developed, as discussed in Chapter 3, is different from others in modelling objects [50], using the face relation graph connecting sensory data, the object geometrical model and grasping features, and could eventually reduce the calculations during data analysis and simplify the object recognition process. Only principal features of objects manipulated are considered.

In order to accomplish the grasping task, it is required to describe an object to be grasped with respect to both its geometry and functionality. A coding system is proposed
for this objective. Many researchers, as reviewed in Chapter 2, focus on the detailed geometric description of objects for assembly. In this study, we emphasize the object's features related to gripping. Based on group technology and an object oriented concept, the grasping oriented coding system with object gripping models is developed for object classification and recognition in application of grasping operations for automated assembly, with a generic robotic gripper.

Object gripping features are represented through the combination of the gripping models and the coding digits. The gripping models are extracted from the object models and the coding digits are generated through a knowledge-based representation. Objects to be grasped are classified based on their gripping models. The coding digits are utilized to describe the object gripping features. The gripping code is automatically generated through the rule-based knowledge inference. In this chapter, the gripping models and coding system are described in detail.

4.2 Object Gripping Models

Object gripping models built in the system are based on the concept of group technology. The group technology, as discussed in Section 2.4, takes advantages of the similarity of parts and processes in design and manufacture, and groups parts into a family in terms of their geometric characteristics and (or) processing requirements. The part code containing sufficient information on part characteristics, can assist a designer to retrieve a design of the part in terms of identical parts, and to find a process planning for manufacture in order to simplify the design and manufacturing process as well as increase the productivity and reduce production costs [3]. Evidently, group technology is application oriented.

Based on the group technology concept, objects with similar grasping characteristics are classified into a same gripping model in order to simplify grasping planning and operation. The object model in the system as discussed in Section 3.4, is used to describe
the features of every part to be grasped in order to perform the recognition task. However, the gripping model is abstraction of the object model, which describes a set of parts with certain common features of gripping. In other words it only extracts the object features related to gripping. Incorporated with the object grasping code discussed in next section, the information on grasping operations is fully provided. Using the gripping model and code, it is possible to establish correspondence and transfer information between the object and the gripper for the performance of the grasping task.

Our purpose is to classify objects of interest into a few families with common attributes related to grasping performance. Therefore, the gripping models are regarded as such families. The grasping features of the objects are fully represented by the models and grasping code. In terms of the principles of group technology and the required grasping performance, the basic gripping models in the system are a block, polyhedron, cylinder, sphere, cone and wedge. In certain particular cases, a complicated object may be described as a combination of some of these models. The establishment of a gripping model for an object to be grasped, is dependent on the properties and relations of the object’s faces to be touched by a gripper. From this point of view, the object-1 and object-2 in Figure 3.10 are both classified as the same “block” gripping model.

The object gripping models extracted features from the object geometrical models, are established according to a feature-based description which can be effectively implemented with a rule-based knowledge inference. An example of such a rule:

IF a pair of parallel faces are found from an object subjected to gripping constraints

THEN its suitable gripping model is a block.

The principal features of the models are faces divided into primary and secondary features in terms of their relation with the operation of the gripper. The face
classification is given in Figure 3.9. The relationship among an image data, its object model and gripping model is depicted in Figure 4.1.

If an image matches a distinguished shape, then it is required to determine what object it is and which gripping family (model) it belongs to. As the matching is successful, the object is recognised and its geometrical data and grasping features are extracted from the data base and knowledge base shown in Figure 4.1. At this stage before the object code digits are generated, the grasping features represented here are face relations such as parallel, cylindrical and so on. These features provide information enough to classify the object into a specific gripping family. Detailed grasping features relative to grasping operations are represented by the coding system discussed in next section.

Gripping models are reasoned by IF-THEN rules associated with forward chaining inferencing [110]. The forward inferencing mechanism is a data driven mechanism with
which the reasoning process starts from observed data, and in principle, proceeds to infer all possible consequences. This means that the data flow in a rule is from the left-hand side of the rule to its right-hand side, i.e. that if the conditions in the left-hand side of the rule is satisfied, then the action in its right-hand side is triggered and taken place. The whole process for inferencing an appropriate gripping model from an image data is taken three steps and may be expressed by IF-THEN rules as follows.

1. IF the image matches a distinguished shape from the distinguished shape feature base
   THEN a specific object is recognised
       (object_name = the recognised object )

2. Extraction of the geometrical data and grasping features of the desired object from the data base and knowledge base in terms of the face relation graph of the object.

3. IF a specific grasping feature is presented with respect to the object AND the feature meets grasping requirements with respect to the gripper designed in the system
   THEN a suitable gripping model is obtained

4.3 Structure of the Coding System

In the proposed system, object grasping features include both geometric and functional characteristics of the object. The geometric features mainly involve the faces types, such as plane or cylindrical faces and faces dimensions. The functional features consist of the relationship among the faces of the object, such as parallel, polyhedral, centroid of the object, and a suitable gripping area. The faces types are described as gripping information embedded in the object model. The faces relationship is obtained through the analysis of a known object described in the object model. Once the object is recognized,
its centroid and gripping area could be obtained based on the previous analysis and image data calculations of the object.

The part coding schemes for part description and classification have been widely used. The Opitz coding system as introduced in Section 2.4, for example, consists of nine basic digits and an extra four digits. The first five digits describe the geometrical attributes of the part, and the other four digits indicate its manufacturing attributes. The extra four digits are intended to represent the production type and sequence. This coding system includes a detailed description of a part and is quite complicate.

Incorporated the object oriented concept with the group technology, however, the coding system proposed in this study is grasping oriented. From the grasping point of view, it is important to identify features related to the grasping operations. Those features, as mentioned above, encompass the object face and its dimensions and centre of gravity, etc.. However, the detailed local shapes of the object are not considered in the coding system as in other coding systems, because they have little relation with the grasping performance under consideration [111]. In the coding system, the two aspects of the geometric and gripping properties of an object are simultaneously considered, which means that the system representing the object to be grasped is dependent on its shape and its attributes related to gripping. The coding system consists of three elements and each one includes several digits to describe the surface properties or gripping attributes respectively. The system is presented briefly in Table 4.1.

<table>
<thead>
<tr>
<th>Coding digit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First element</td>
<td>Primary features of an object</td>
</tr>
<tr>
<td>Second element</td>
<td>Face types to be grasped</td>
</tr>
<tr>
<td>Third element</td>
<td>Gripping attributes</td>
</tr>
</tbody>
</table>

Table 4.1 Proposed coding system structure
In the first element, there are six digits (1, 2, ..., 6) used to describe the primary features of the six gripping models, respectively. These six digits actually classify objects into six different gripping families. The second element is indicative of the face types. For example, the gripping model of a block includes a pair of faces for gripping which may be composed of two planar faces or one planar and one curved faces. Therefore, the model of the block has a wider significance than the geometrical description. This extended meaning of the gripping models is one of the characteristics of our system, which makes it possible to involve more parts with gripping similarities. The third element describes (1) the maximum distance between any two grasped faces of an object by digits, which is in the range of the specific gripper designed for the proposed system, and (2) the faces touch points by the tips of the gripper, which determine both the final grasping position and the approach direction and distance of the gripper when the grasping operation is carried out. The touch points are defined as the distance from the top surface of the object to the points, when an object is fed to the scene. To represent the third element of the code, two digits are required to describe the gripping attributes. For example; the object-1 in Figure 3.10 has code 1-1-2-2. Where, the first digit 1 indicates that primary feature of the object is the block with two parallel faces; the second digit 1 means that the pair of faces for grasping are planar - planar; the third digit 2 describes that the distance between the central point of the model and one of the grasped faces is about 10 ~ 20 mm; the last digit 2 indicates that the approach distance of the gripper is also about 10 ~ 20 mm. The detailed description of the coding digits is given in Table 4.2 and Table 4.3.

<table>
<thead>
<tr>
<th>Primary Features</th>
<th>Block</th>
<th>Polyhedron</th>
<th>Cylinder</th>
<th>Sphere</th>
<th>Cone</th>
<th>Wedge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digit</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 4.2 Coding digits for the primary features
### Table 4.3 Coding digits for the face types and gripping attributes

<table>
<thead>
<tr>
<th>Face Types</th>
<th>Digit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planar-Planar</td>
<td>1</td>
</tr>
<tr>
<td>Planar-Curved</td>
<td>2</td>
</tr>
<tr>
<td>Planar-Wedged</td>
<td>3</td>
</tr>
<tr>
<td>Planar-Planar-Planar</td>
<td>4</td>
</tr>
<tr>
<td>Cylindrical</td>
<td>5</td>
</tr>
<tr>
<td>Spherical</td>
<td>6</td>
</tr>
<tr>
<td>Conical</td>
<td>7</td>
</tr>
</tbody>
</table>

#### Gripping Attributes

<table>
<thead>
<tr>
<th>Maximum Distance between Faces (mm)</th>
<th>Approach Distance (mm) (Touch Position)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digit</td>
<td>Digit</td>
</tr>
<tr>
<td>&lt;10</td>
<td>&lt;10</td>
</tr>
<tr>
<td>11 - 20</td>
<td>11 - 20</td>
</tr>
<tr>
<td>21 - 30</td>
<td>20 - 30</td>
</tr>
<tr>
<td>31 - 40</td>
<td>31 - 40</td>
</tr>
<tr>
<td>&gt;41</td>
<td>&gt;41</td>
</tr>
</tbody>
</table>

It should be noted that in the planar-curved pair, the curved face is any curved face except the cylindrical, spherical and conical faces and the planar-planar-planar faces imply that the three faces extracted from a polyhedron. And the maximum distance between the grasped faces are measured by the distance from the centre of these grasping elements to one of these faces which gives a maximum value by measurement. The maximum range of the gripper can reach is limited to be 85mm in size and 50mm in depth (approach direction). The coding system is constructed as the chain-type structure. Each digit is independent of others and it represents each attribute in each position fixed in the sequence of the code.

#### 4.4 Knowledge Based Code Generation

In most available coding system such as the Opitz coding system [66] as reviewed in Section 2.4, the part code is generated manually, which heavily loads on the designer’s preparation time. Many researcher have investigated the automation of code generation
It is possible to generate the code automatically by means of an expert system or a neural network system. For example, the system CODER developed by Henderson [112] is a rule based program using CAD data to automatically generate a part code. CAD data of the part are extracted by a feature recognizer, and then the part code is generated through a feature interpreter based on the production rules. Kaparthi and Suresh [113] investigated a neural network system to generate part geometry-related digits of the Opitz code. The system is a multi-layer network trained by a back propagation algorithm. The input of the system is bimaps of part drawings and the system output has ten serial values which are either 1 or 0 relative to ten digits of the Opitz code. When the output is 1 in a digit position, it means the attribute of the part is belong to this class. Similarly, the Opitz code could be automatically generated. One of the important issues of generating the part code automatically, is to identify the object to be grasped in the scene and extract its features for the code generation.

In our system, when an object is recognized in the matching procedure in terms of feature extraction and model description, it is classified into a specific family with respect to the gripping models. At the same time, the code for description of its grasping features is produced in terms of a knowledge based code generation approach. Figure 4.2 is an outline of the knowledge based system for automatic generation of the code's digits.
The knowledge based system in Figure 4.2 illustrates a decision process, from bottom to top, on classification and description of an object by a coding system with respect to an appropriate gripping model of the object. The determination of object models is of key importance in the whole system, which has a great influence on the decision of each digit of the object as well as the relevant gripper configuration. When the grasping features of the object are extracted by recognition and a suitable gripping model is then obtained by inference, these features are checked along the decision process and interpreted to the code digits. All information on the object to be grasped is represented through its code as well as its gripping model which integrates with the code as a whole. Therefore, the object is recognised and classified by its principal features relative to gripping.

Two aspects of knowledge of an object are required for the code generation: geometrical characteristics (face types and dimensions) and functional characteristics (face relations in gripping, location and orientation in the scene). The relationship of the object grasping features and its code digits is represented by production rules in a knowledge base. The
code generation is hierarchically proceeded in three steps with relation to the three elements of the coding system, as see in Figure 4.2.

A production rule system offers a type of formalism for representing knowledge, in which the actual problem-solving knowledge is effectively expressed in terms of rules [114] [115]. The expression of domain knowledge takes the form of

\[
\text{IF} \quad \text{certain conditions are fulfilled} \quad \text{THEN} \quad \text{certain actions are taking place}
\]

The conditions are generally composed of AND or OR relations. The actions (or conclusions) may encompass several parts and have a logic or heuristic relations with the conditions. For example, here is a heuristic rule:

\[
\text{IF} \quad \text{There are two parallel surfaces of an object} \\
\text{AND} \quad \text{the surfaces satisfy the constraints of gripping} \\
\text{THEN} \quad \text{Its gripping model is a block} \\
\text{AND} \quad \text{the digit of the first element of the object code is 1}
\]

The conditions consist of facts which represent domain knowledge about objects in geometry and gripping. These facts are stated corresponding to the description of object models. For example of the object-1 in Figure 3.10, there may be some fact statements as follows.

Object Name = Block-1

Face1_Face2 = Parallel_Planar_Planar
Face3_Face4 = Parallel_Planar_Planar
Dis_Face1_Face2 = 30 (mm)
Dis_Face3_Face4 = 50 (mm)
These facts above indicate the object name, its faces types (planar) and relations (parallel), and the distances of the two pairs of faces, which represent the grasping knowledge of the object. The grasping domain knowledge represented above seems frame-like in its structure [114] [115] which acts as one part of the conditions of the production rule. The other part of the conditions is provided by checking if the two pairs of faces meet the gripping constraints, and then a decision can be made which pair is better for gripping as grasping elements. Once all conditions are correctly provided, the rule is going to fire i.e. the action part is taken place and a appropriate code digit is generated. Based on the production rules with forward chaining inference, as already mentioned and shown in Figure 4.2, the reasoning process of code generation follows three steps and discussed in detail as follows.

4.4.1 Coding for Primary Features and their Relations

As aforementioned, primary features and their relations with respect to gripping models, are of those properties that determine which family the object belongs to, and in the interim, are selected as the grasped elements. For example; if two parallel faces (at least one of them planar) are identified in comparison with the gripping models, the object is classified into the block family; if a cylindrical face is extracted, the object is classified into the cylinder family. Then they are coded by digits. In practice, when an object is presented in the scene by a feeder, only side surfaces of the object are usually considered as the potential gripping faces. Before the object is grasped, all side faces are checked through a serial of filters in order that the selected gripping faces meet the grasping and assembly requirements [79] [111]. These filters are regarded as constraints under the gripping environment. Failure to pass the filters, means that the object is not appropriately presented in the scene to be grasped. Thus it is rejected by the system.

The purpose of filtering is to select the grasping elements from several grasping features’ candidates. The filtering process takes into account the object geometry, the gripper structure, the gripping attributes and the assembly or other processing tasks, and
incorporates all the aspects together. For the assembly tasks, the mating faces of an object in assembly could not be selected for gripping. A number of filters are utilised in the system. For instance, filters for a parallel pair are as follows.

Filter 1. Neither of the faces in a pair is a mating face. In the meantime, the access of the mating part is checked when the gripper is preparing to perform the grasping and assembly task. This filter makes it possible to achieve the assembly task.

Filter 2. The distance between the two faces is less than the maximum distance between fingers when applied to grasping. It ensures the gripper dimensional capability for grasping the object.

Filter 3. A pair with a smaller distance is preferred.

Filter 4. A pair overlapping the centre of mass of the object is preferred.

Filter 5. A face with a big area of contact in gripping is preferred. This filter imply that a planar face is superior to a curved face in gripping.

Filter 6. A pair with convex faces is preferred. This filter may prevent the gripper from colliding with the object when the gripper is approaching to the object.

The mating face of an object is determined by analysing the known object and attached to the object model beforehand. For simplicity in the system, it is assumed that all the side faces are not mating faces if the object is properly placed in the scene. In the filter 3, the difference among the distance of each pair under filtering is estimated by a specified tolerance $T_{\text{Dis}}$. Currently the $T_{\text{Dis}}$ is set 10mm. That is to say that if the tolerance is more than 10mm when two pairs are compared each other, the pair with larger distance is rejected. Otherwise, both of those two pass the filter. If the pair overlaps the centre of mass of the object is also determined beforehand. The convex characteristic of a face is described with its face type description. All these filters are not given a same weight in checking face pairs. Their precedence is given in Figure 4.3.
Filter 1 and Filter 2

↓

Filter 3, Filter 4, Filter 5 and Filter 6

**Figure 4.3** The precedence of filters

The precedence indicates that if a pair does not pass the filter 1 and filter 2, it is certainly rejected. For the rest filters, the best candidate is selected according to evaluation scores. If a pair passes a filter, it gets score 1, otherwise 0. The winner is one who gets highest score of all candidates. For cylindrical, conical and spherical faces, usually only the filter 1 and filter 2 are employed. The filter 4 and filter 6 are supposed to be automatically satisfied. In these applications, three fingers are required to carry out the grasping operations with respect to the specific gripper designed in the system. Grasping a polyhedron also needs three fingers of the gripper.

The filtering process in gripping code generation is also considered as a reasoning process. The filtering is represented as a procedure which is described in Figure 4.4.

**Filtering procedure:**

1. Check a pair by filter 1 and filter 2.
2. If failure, the pair is rejected. Choose another pair and goto step 1.
   - If success, goto step 3.
3. Check the pair by next filter.
4. Initialise score = 0.
   - If success, score = score + 1; if failure, score = score + 0.
   - Goto step 3 till all filters are employed.
5. Save the score with the pair.
6. Choose another pair and repeat step 1 to step 6 till all candidate pairs are checked.

**Figure 4.4** Filtering procedure
The successful pair with the maximum score is selected as the grasping elements. Up to now, the conditions of the heuristic rule mentioned in the beginning of this section, for inferencing the object model and producing the primary feature digit, are fully provided and the rule is going to fire. Actually the grasping features of an object such as ‘parallel’ are connected with the distinguished shape of the object through the face relation graph. As soon as the distinguished shape is matched, the gripping features and geometrical data of the object are extracted and attached with the object’s name, refer to those facts statements mentioned before. Suppose that the object-1 in Figure 3.10 is given name as Block-1 and placed in the scene. After the processes of recognition and filtering, the heuristic rule mentioned above can be detailed and simplified as

\[
\text{IF } \text{Objrct\_Name} = \text{Block}-1 \\
\text{AND } \text{Pass\_Filter} = True \\
\text{THEN } \text{Gripping\_Model} = \text{Block} \\
\text{AND } \text{Object\_Code1} = 1
\]

The object name ‘Block-1’ implies that the object has parallel faces, and the ‘Pass\_Filter= True’ implies that the face 1 and face 2 pair passed the filtering process with a highest score and will be as the grasping elements for further grasping performance.

4.4.2 Coding for Face Types of Gripping

Face types are classified shown in Figure 3.9. In many cases, the significance of face types are same as that of primary features mentioned above. For example, a cylinder family should have a cylindrical face type. However, a block family may include two planar faces or a planar and a curved faces. The faces combined as grasping elements and their corresponding digits are shown in Table 4.3. How many faces are required for grasping an object are determined in terms of its gripping model and with respect to the gripper designed in the system. Usually, for instance, for a block at least two fingers are
needed to touch the two parallel faces of the block; for a cylinder it is better to use three fingers to touch three parts of its cylindrical face which may be assumed to be composed of three cylindrical faces in gripping.

Face types are of importance in determination of gripper operations. For our specific designed gripper with three fingers referred to Figure 7.2, a different joint motion is required by a plumb plane and a wedged one to be grasped. The identification of face types is made by the incorporation of the shape identification and the object description. Once the object is recognised, their faces types are extracted and identified, and then they are interpreted into the code digits. Based on the face types encoded with the digits, which joints of the gripper to be actuated are thereafter determined.

4.4.3 Coding for Gripping Attributes

Gripping attributes include the dimension, location and orientation of the object as well as the grasped position of its faces. The dimension here especially indicates the distance of the grasped faces. For simplicity, we only code the dimension and the grasped position of faces because they have direct relations with the gripper's performance. The location and orientation of the object are represented with the gripping model.

The distance of the faces to be grasped is measured from their geometric central point to the faces. For two parallel faces, it is a half of the distance between the two faces; for a cylindrical face, it is its radius. The grasped position of the faces is, from the approach direction of the gripper point of view, defined as the distance from the grasped face edge to the grasped point that is desired to be in the centroid of the object as close as possible. Their values are obtained from the analysis of the object surfaces and coded as digits in terms of different value ranges, refer to Table 4.3.

The coding for face types and gripping attributes are also represented by IF-THEN rules.
Following the primary feature of the object-1 shown in Figure 3.10 encoded, the following rules are actuated:

\[
\begin{align*}
\text{IF} & \quad \text{Face}_1 \_\text{Face}_2 = \text{Paralle}_\text{Planar} \_\text{Planar} \\
\text{THEN} & \quad \text{Object\_Code}_2 = 1 \\
\text{IF} & \quad (\text{Dis}_\text{Face}_1 \_\text{Face}_2) / 2 \geq 11\text{mm} \\
& \quad \text{AND} \quad (\text{Dis}_\text{Face}_1 \_\text{Face}_2) / 2 < 20\text{mm} \\
\text{THEN} & \quad \text{Object\_Code}_3 = 2 \\
\text{IF} & \quad (\text{Object\_Height}) / 2 \geq 11\text{mm} \\
& \quad \text{AND} \quad (\text{Object\_Height}) / 2 < 20\text{mm} \\
\text{THEN} & \quad \text{Object\_Code}_4 = 2
\end{align*}
\]

The value of the object height is extracted from the object model as the object is recognised. Object height values are also stated as facts for providing rule conditions to fire an appropriate rule.

4.5 Conclusion

The coding system concept presented in the chapter is effectively used for object representation and classification in the application of grasping operations. Objects to be grasped are classified and grouped into certain families in terms of their gripping models. The gripping models catch the major characteristics of the objects and abstract them as models for gripping. It has certain advantages to classify objects according to their gripping models:

1. Object can be directly connected with grasping;
2. The geometrical information may be easily transferred to grasping and assembly with the models; and
3. The reasoning process in the system may be simplified by means of the gripping models which link geometrical, grasping and assembly information.

The object gripping features are encoded by means of the coding digits which constitute a chain type structure, based on object gripping models. The coded grasping information is easily extracted and transferred when it is needed. The coding process is a knowledge reasoning process in which the relations of the grasping features and the corresponding digits are represented by IF-THEN rules. The reasoning of digits is hierarchically performed with respect to the three levels of coding elements. Therefore the gripping code is automatically generated through the rule-based knowledge inference.
5.1 Introduction

The coding system discussed in Chapter 4, is applied to describe the gripping features of the object to be grasped. In order to accomplish the grasping task, the object gripping code should be transferred to a series of operation commands for a gripper. The transformation approaches and procedures are required to be simple and effective as well as be carried out on line.

Usually, since different objects have different geometrical shapes, different grasping operations are required to be performed with respect to these different shapes. There could be a great number of operations taken for a few objects. As similar objects are classified into a same family in terms of their common grasping features in our system, the corresponding gripper operations would be decreased and simplified. In a same family of objects, similar gripper configurations are given and similar operations are performed. The information on an object stored with its code and the corresponding gripping model is conveyed to the gripper in order to establish the correspondence between the object data and the operation data and perform the grasping tasks. In the system, an explicit method using specific matrices to represent the information and establish the correspondence (matching) between the gripper and the grasped faces is utilised. The object gripping information and the gripper configuration will then correspond.

In this chapter, two matrices: the object matrix and the gripper configuration matrix, to interpret the gripping information for establishment of the correspondence of object gripping features and operations are presented. A knowledge-based structure to match these two matrices is illustrated.
5.2 Object Matrix

The proposed gripper is designed with three gripping fingers, therefore requiring maximum three object faces to be selected as the grasped parts. In a multiple finger gripper design, same principles could be applied. The face number, the face type, the distance between faces and the centre of the object, and the contact point are selected as the elements of the matrix. Thus the object matrix has a form shown in Figure 5.1.

<table>
<thead>
<tr>
<th>Face No1</th>
<th>Face No2</th>
<th>Face No3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected face</td>
<td>$a_{11}$</td>
<td>$a_{12}$</td>
</tr>
<tr>
<td>Face type</td>
<td>$a_{21}$</td>
<td>$a_{22}$</td>
</tr>
<tr>
<td>Distance between faces and the centre of the object</td>
<td>$a_{31}$</td>
<td>$a_{32}$</td>
</tr>
<tr>
<td>Contact point of faces</td>
<td>$a_{41}$</td>
<td>$a_{42}$</td>
</tr>
</tbody>
</table>

Figure 5.1 Typical form of the object matrix

Where: $a_{ij}$ indicates selected faces. If a surface is selected as a gripping face, the $a_{ij}$ takes 1. otherwise it is 0.

$a_{2j}$ indicates a face type. There are several face types and corresponding $a_{2j}$ values.

$a_{3j}$ indicates the distance between faces and the centre of the object. The values of $a_{3j}$ are dependent on the actual distance range that is defined as a very small, small, medium, large and very large one. The actual distances and the corresponding $a_{3j}$ values were established. $a_{3j}$ proximately describes the size of the object in two dimensions.
$a_{4j}$ indicates the contact point of a face with the tip of a finger. The point is measured in terms of the distance between the point and the face edge where the gripper is supposed to approach in. Depending on its accuracy, the distance is divided into five ranges. Detailed values for $a_{2j}$, $a_{3j}$ and $a_{4j}$ are predefined by the proposed system (see Appendix A).

The matrix collects and digitises the object features related to gripping in terms of the information of an object code and its gripping model. The relationship between the matrix elements and the object features is described in a knowledge base system. When the object code and its gripping model are obtained, the matrix is built through an algorithm based on the knowledge representation in the knowledge base. The knowledge is represented in a form of IF - THEN statement. For example, the following rule could appear in creating a matrix:

$$\text{IF } \text{the face type is planar, }$$
$$\text{THEN } \text{the value } a_{2j} \text{ is 1.}$$

The code of the object in Figure 3.10, as aforementioned, is 1-1-2-2. Where, the first digit 1 indicates that the primary feature of the object is a block with parallel faces; the second digit 1 means that its all two faces for gripping are planar; the third digit 2 describes the maximum distance between the faces and the centre of the object. And the last digit 2 gives the approach distance of the gripper. Obviously, its corresponding gripping model is a block. The object matrix is created while the meanings of its values are predefined in the system. Figure 5.2 shows the process of how the code and object model is transformed into the matrix by means of incorporation of a tree search algorithm and knowledge inference.
The elements of the object matrix, as shown in Figure 5.2, are obtained in terms of the object code and its gripping model. This process is performed in three steps as follows.

1. Decomposing the object code digits. (Since the code digits are represented as a chain-type structure, it is easy to decompose it and extract its information embedded in the digit.)
2. According to each digit decomposed, each specific production rule is searched in the knowledge base.
3. If the condition of a rule is satisfied, the rule is invoked and fired. Then a matrix element is obtained by the interpretation of the action part of the rule.
The process of the matrix generation is hierarchically carried out. The elements \(a_{ij}\) are produced based on the object gripping model and its primary feature i.e. the first digit of an object code. Different gripping models require different number of faces to be touched by the gripper; the \(a_{2j}\) on the second digit of the code; the \(a_{3j}\) on the third digit of the code; and the \(a_{4j}\) on the forth digit of the code. For the object-1 in Figure 3.10, its matrix can be represented as follows in terms of the three steps above.

\[
\begin{bmatrix}
1 & 1 & 0 \\
1 & 1 & 0 \\
2 & 2 & 0 \\
2 & 2 & 0 \\
\end{bmatrix}
\]

The meanings of its values are referred to Appendix A.

5.3 Gripper Configuration Matrix

The gripper configuration (position and orientation of its fingers) is defined as a configuration where the gripper's fingers are going to touch the grasped faces of the object, i.e. the destination state of the gripper required to begin to perform its grasp operation. The destination is the initialisation point of the gripper for gripping. In order to match the object matrix, the gripper configuration matrix should have identical elements with the object matrix. Obviously, the gripper configuration is formed by all its fingers configurations which are based on the tips positions of the fingers. The gripper matrix has a form shown in Figure 5.3.
<table>
<thead>
<tr>
<th>Selected finger</th>
<th>( b_{11} )</th>
<th>( b_{12} )</th>
<th>( b_{13} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finger configuration</td>
<td>( b_{21} )</td>
<td>( b_{22} )</td>
<td>( b_{23} )</td>
</tr>
<tr>
<td>Finger position relative to the centre of the gripper</td>
<td>( b_{31} )</td>
<td>( b_{32} )</td>
<td>( b_{33} )</td>
</tr>
<tr>
<td>Tip position</td>
<td>( b_{41} )</td>
<td>( b_{42} )</td>
<td>( b_{43} )</td>
</tr>
</tbody>
</table>

**Figure 5.3** Typical form of the gripper configuration matrix

Where: \( b_{1j} \) indicates selected fingers. If a finger is selected for grasping, the \( b_{1j} \) takes 1. otherwise it is 0.

\( b_{2j} \) indicates a finger configuration. It has a close relation with a face type on which the finger should touch in grasping. There are several face types and corresponding predefined \( b_{2j} \) values. A finger configuration implies which joints of the finger will be operated for grasping.

\( b_{3j} \) indicates the finger position relative to the centre of the gripper when a grasping operation is to be performed. The finger position is measured in terms of the distance between a finger (here means its joint A, refer to Figure 7.2) and the centre of the gripper. The distance is divided into several ranges and corresponding \( b_{3j} \) values were selected. The distance is defined here according to a relative position of a finger.

\( b_{4j} \) indicates the tip position of a finger, which corresponds with the contact point of the face to be grasped. The tip position is measured in terms of the distance between the contact point of the face and the face edge where the finger tip starts approaching into
the face. Its ranges and corresponding $b_{s_j}$ values were determined. Specific data related to values of $b_{2_j}$, $b_{3_j}$ and $b_{4_j}$ are predefined by the system, refer to Appendix B.

In comparison with the related object matrix, the gripper configuration matrix has the following distinct characteristics: It has same size consisting of $4 \times 3$ elements; It has same value for same matrix element; Same values have different meanings for same elements.

The matrix is built in terms of the corresponding object matrix and a knowledge base stored in the system. The knowledge base includes two kinds of information: static and dynamic. The static knowledge represents the gripper size, its possible grasping range, and its initial position, and the dynamic one represents the relationship between the gripper matrix elements and its configuration dependent on the object features.

Relative to the object to be grasped, the gripper matrix represents the initial state of the gripper to carry on grasping, the operations to be performed and the joint movements to be taken. In the same family of objects, the relevant gripper configurations and operations are similar, but only the range of their joint movements are not same which depends on the different object’s size. Based on group technology and the coding system proposed, the gripper operations (which joints are going to be actuated.) are classified corresponding to the object gripping models.

5.4 Matching of the Object and Gripper Matrices

The matrix matching means that the information of an object has been transformed into the data for the gripper position for gripping. Virtually, the purpose of the matching is to choose a suitable gripper configuration to fit the object to be grasped. The information flows from the object matrix to the gripper matrix, but the matching process is carried out from the gripper to the object through knowledge inference, shown in Figure 5.4. A
series of matching rules are fired by the information in the object matrix to proceed the matching performance.

We divide the matching process into two steps. One step is to establish the correspondence between the two matrices and the other is to derive a geometrical transformation for the determination of the gripping position of the gripper in terms of this correspondence. Here only the first step is detailed. The geometrical transformation will be explained in Chapter 6.

5.4.1 Correspondence of the Two Matrices

The correspondence of the information between an object and the gripper configuration in the matrices is obtained according to both the representation of information and the search algorithm. The IF-THEN production rule based representation and inference is utilised in the system.
The corresponding process for matching is hierarchically divided into three stages in Figure 5.5. Firstly the correspondence of information of row 1 in the two matrices is established through a search algorithm, then row 2 between them. Lastly, the row 3 and row 4 of the matrices are corresponded.

![Diagram showing the matching stages]

First of all, the finger - face pairs are determined. In the matching stages, the selection of the finger for a given face, is considered as a primary step in matching. Generally, for the three fingers to fit three faces respectively, there are 3x2 possible combinations. We use $s_i$ ($i = 3$) to denote the faces of the object and $f_i$ ($i = 3$) the fingers of the gripper. Thus, in the state space of the problem, we get

\[
\text{Initial state: } (s_1, s_2, s_3) \cap (f_1, f_2, f_3)
\]

\[
\text{Goal state: } (s_i, f_i) \text{ pairs.}
\]

Obviously, the problem to be solved is how to get the goal state, i.e. to select appropriate face - finger pairs. The search tree in solution of the problem is built in Figure 5.6.
Here, every path from its root to terminal leaves is one possible solution of the problem. The six possible solutions are

\[
\{ (s_1, f_1), (s_2, f_2), (s_3, f_3) \} \\
\{ (s_1, f_1), (s_3, f_2), (s_2, f_3) \} \\
\{ (s_2, f_1), (s_1, f_2), (s_3, f_3) \} \\
\{ (s_2, f_1), (s_3, f_2), (s_1, f_3) \} \\
\{ (s_3, f_1), (s_1, f_2), (s_2, f_3) \} \\
\{ (s_3, f_1), (s_2, f_2), (s_1, f_3) \} \\
\] (5.2)

In order to get better solution in every grasp, actually there are geometrical constraints imposing on the search in terms of the object gripping features and gripper structure. It is possible to arrange the three fingers of the gripper as finger 1, 2, and 3, and to take the finger 1 and finger 2 as major grasping parts and the finger 3 the auxiliary part.
beforehand. Thus the matching search will be much simplified, especially in the case of matching a block family. For most cases in the block family, only those two major fingers are taken into account and applied in gripping.

The purpose of the finger configuration to match the face type is to determine which joints of the gripper will be operated in grasping the object in the scene. The determination of joint movements relates to the gripping models and the object code. For the specific gripper consisting of three fingers each of which has three joints, refer to Figure 7.2, if two fingers are used to grasp a block, in principle, only the joint A is requested to operate. To grasp a cylinder, both the joint A and joint B are required to manipulate. For a wedged or conical face to be grasped, the joint C will certainly be operated. In this stage of matching, the joints to be operated are not explicitly represented in the gripper matrix and will be interpreted in the form of gripper code digits based on knowledge representation in the next chapter in order to generate the detailed operation commands to manipulate the gripper.

The matrix elements $a_{3j}$, $a_{4j}$ and $b_{3j}$, $b_{4j}$ describe not only the gripping attributes but also the object face positions which the gripper should approach to. In order to match the finger position with the face position, it is necessary to determine the kinematic movement of the gripper. This process may be divided into two steps. The first step is to move the gripper from its origin to its destination, the location of the object to be grasped. And the second step is to manipulate its fingers to the desired positions to start its grasping operations. The detailed manipulation embedded with the values of the matrix elements will be discussed in Chapter 6.

The correspondence may be established for the object and gripper matrices through the three matching stages mentioned above. The search algorithm for the matching, as aforementioned, is based on knowledge inference. Generally speaking, the search algorithm is expressed as the following Procedure.
Procedure search

1. Put state in initial state.
2. While state ≠ goal state do.
4. Select PRODUCTION RULE applicable to state and fire this (or these) RULE (s).
5. State := new state (produced by acting the RULE).
6. End.

5.4.2 Knowledge Representation

The whole system developed here, from the image collection, object recognition to the code generation and matrix formation as well as the generation of operation commands in next chapter, is knowledge based. The knowledge representation is concerned with knowledge of the images, the objects and the gripper as well as the processes of object recognition, code generation, gripper operation and so on. The domain knowledge in this application may be catalogued as the static knowledge and the dynamic one. In this section, a general knowledge representation and inference is presented and the knowledge base structure for establishment of correspondence between the object and gripper matrices is described.

5.4.2.1 Static and Dynamic Knowledge

The static knowledge may be intrinsic, unchangeable with respect to its environment. In most situations it may be expressed in declarative formalism. This kind of knowledge in the system involves the image (as long as being collected from the sensory system, they are not changed), the object and the gripper. For the image the static knowledge includes the size, the value at each point (grey level) of the image array. For the object it deals with the dimension, the faces, the faces types and their relations as well as the weight of the object. For the gripper it involves the size and structure, the number of the fingers and joints, the motion types of the joints of the gripper. The static knowledge acts as
fundamental elements for construction of the whole system proposed. However, the image, the object and the gripper will have no any relationship without the help of the dynamic knowledge.

The dynamic knowledge may be logical, heuristic, changeable under environment and could be expressed in procedural style. It establishes the relations among the image, the object and the gripper. The dynamic knowledge in the system comprises the following aspects.

- knowledge for extraction of the image features
- knowledge for building the face relation graph and the object models for object recognition
- knowledge for description of the object gripping features and models and generation of the coding system
- knowledge for generation and matching of the object matrix and the gripper matrix
- knowledge for generation of the gripper code and operation commands

The dynamic knowledge above acts as procedures to connect different domain knowledge and to reason them for problem solving. These are illustrated in the relevant chapters.

5.4.2.2 Rule-Based Representation and Reasoning

As aforementioned, knowledge, both static and dynamic, is mixed in the IF-THEN rule-based representation. The production rule formalism can support both declarative or procedural knowledge. It explicitly represents human domain knowledge and easily forms a data-driven forward chaining inference. Normally, a production system is composed of a working memory, a knowledge base (rule base) and an interpreter (inference engine) [116]. In the system developed, the working memory, rule base and interpreter are neither explicitly nor respectively set up as a whole. According to task
decomposition, the domain knowledge is decomposed into certain specific areas such as object recognition and code generation, and the area knowledge is represented in a rule base (we may still call it like this) which is separately distributed and involved in each reasoning procedure. A general reasoning process may follow three steps:

1. to select Candidate Objects with their environments to be reasoned. This step is to find out the reasoned objects and sufficient data to support the reasoning.
2. to analyse the Candidate Objects and catch their desired features for reasoning. This step is to determine which features of the Candidate Objects are related to the problem solving.
3. to analyse and infer the predicted outcome with respect to the features. This step is a decision making for achieving the reasoning goal.

For the matrix matching of objects and the gripper regarding the reasoning steps above, the Candidate Objects are faces and fingers and the features of these Candidate Objects are the desired number of faces and fingers for gripping, face types and joint motion styles, as well as face and finger positions. The reasoning goal is to match the fingers with the faces in order to transform the object gripping information into the gripper.

Reasoning procedures are written by Visual Basic (VB), an object-based, event-driven programming language [117]. Different area knowledge is represented in different procedures and common knowledge and processes are shared by relevant procedures through the global definition of VB. Such a structure for a small and medium size of knowledge representation has following advantages:

1. It is easy to find out an appropriate rule to reason and solve a desired problem so as to avoid rule conflict in firing. Some conflict resolution strategies may be found in [115].
2. The searching space in reasoning is limited with respect to a specific task.
3. The data-driven forward chaining is easily adopted.
4. The programming structure is simple and clear.
5. It is suitable and easy to implement the system with the VB program.

From a systematic point of view, the knowledge base in the system is composed of four levels which form a hierarchical structure shown in Figure 5.7.

![Knowledge base consisting of four levels](image)

How to apply an object oriented program language such as C++ to describe the coding and grasping system and systematically reason about the matching and gripping is referred to Katz and Huang [118]. More detailed knowledge representation using the object orientation structure may be referred to Czejdo *et al* [119] and Bahr *et al* [120].

5.4.2.3 Knowledge Representation for Matrices Matching

The matching of the object and gripper matrices is made in the matrix generation level. As shown in Figure 5.5, it takes three steps to perform the matching. There are two kinds of knowledge provided for the matching. One is for the representation of the information in the object matrix and the other is for the information of the gripper configuration. These kinds of *a priori* knowledge is represented by IF - THEN rules in
order to match a suitable gripper configuration with the object to be grasped. The knowledge representation is designed with a hierarchical structure which comprises several layers of information (Figure 5.8), relative to the matching stages shown in Figure 5.5.

There are four layers in use as follows:

1. Fingers to be selected relative to the faces selected for gripping;
2. Joints to be actuated relative to the faces types;
3. Joints movements relative to the face positions; and
4. Gripper approaching relative to the face positions.

The search is limited by being carried out at the same layer of the knowledge representation, which substantially decreases the searching and reasoning time. Once the object matrix is formed based on the object code and its gripping model, the searching starts, from the object knowledge representation. When the corresponding features at all layers of the representation are searched and reasoned, the gripper configuration matrix
is formed and at the same time, the correspondence between the two matrices is established.

5.5 Conclusion

A matrix representation for transformation of the grasping information between the object and the gripper is described and the methodology of knowledge representation and reasoning for the study generally discussed in this chapter. The object matrix details the information provided by coding digits and describes the grasping features with respect to the object faces to be touched by the gripper fingers. The gripper matrix represents the gripper configuration and fingers positions desired for gripping. The gripper configuration is determined by each finger configuration which is further described by the finger position. With the matrix representation, the gripping information can be transferred from objects to the gripper and the correspondence of the gripper and the objects to be grasped be established.

Based on the matrix representation, an IF-THEN rule-based knowledge representation and reasoning is provided for matching the gripper matrix with the object matrix. The matching is carried out in three layers with respect to the grasping information represented with the matrices rows. As soon as the matching is done, a gripper configuration is formed corresponding to the object to be grasped in the scene and the desired fingers positions are determined. Accordingly the correspondence of the two matrices is established.

In general the IF-THEN rule representation is straightforward and also corresponds to human processing. In our system, there are four levels of knowledge reasoning processes (Figure 5.7) with respect to four major events. These events are hierarchically taken place from up to down. Thus a data-driven forward chaining is adopted for the system. The data flow is guided by the event-driven control with respect to grasping information.
CHAPTER 6 GRIPPER CODE AND GRIPPER OPERATIONS

6.1 Introduction

The gripper configuration matrix discussed in last chapter represents the gripper information relative to the object to be grasped. All the major gripping features of the gripper are embedded with the values of the matrix elements. Therefore, by means of the matrix, the desired operations of the gripper is properly determined. However, the gripper operations are only implicitly described with the matrix elements. It is necessary to explicitly indicate which joints should be operated in gripping and how much movement be taken, and then to generate appropriate operation commands to manipulate these joints and drive them to their desired positions in the space. As a result, the gripping information embedded in the gripper matrix is materialised and a proper grasp is achieved finally.

In this chapter, a gripper code is proposed to describe the joints to be operated in gripping. The code acts as a means to detail the gripper configuration matrix and convey the gripping information to operation commands. A series of operation commands of the gripper generated with respect to the gripper code are discussed. A coordinate transformation from the gripper position to the object position and a circuit for control of the finger are briefly presented.

6.2 Gripper Code

Gripper fingers are driven by a series of commands which are represented by a gripper code in the system. Its aim is to manipulate the gripper towards the accomplishment of the rough grasp. The fine grasp task is accomplished when the gripper touches the object and a sensory feedback system is operated. This function of the gripper code is a part of
The gripper code system is based on its configuration matrix. A knowledge-based inference (in the form of IF-THEN rules) is taken for generating the code digits in terms of the matrix. Such the gripper code would have a close relation with its joint movement that determines the finger configuration. The code system consists of several elements which form hierarchically a three-layer structure (Table 6.1).

<table>
<thead>
<tr>
<th>Coding digit</th>
<th>Description (Joint movement)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finger 1</td>
<td>Joint A</td>
</tr>
<tr>
<td>Finger 2</td>
<td>Joint A</td>
</tr>
<tr>
<td>Finger 3</td>
<td>Joint A</td>
</tr>
<tr>
<td>First layer</td>
<td>Joint A</td>
</tr>
<tr>
<td>Second layer</td>
<td>Joint B</td>
</tr>
<tr>
<td>Third layer</td>
<td>Joint C</td>
</tr>
</tbody>
</table>

Table 6.1 Proposed gripper code system

The coding system as implemented, performs the following: fingers number and actual fingers are determined. The finger configuration according to the gripper design is described. The number of joints and the actual joints to be operated in gripping are selected. A finger position is determined by its joints. The position and orientation of the gripper relative to the centre and principal line of the object is determined by the geometrical transformation. In principle, the coding system describes the terminal position of the gripper for gripping. The detailed description of the relations between code digits and joint movements is given in Appendix C.

The three joints of the gripper perform three different types of motions, refer to Figure 7.2. The joint A performs linear movement; the joint B rotation and the joint C yaw. The
The three joints of the gripper perform three different types of motions, refer to Figure 7.2. The joint A performs linear movement; the joint B rotation and the joint C yaw. The combination of these joint movements forms a gripping configuration with respect to the object to be grasped. The digital number of the code represents the actual movements of joints for a rough grasp. In order to achieve the rough grasp, it is required that the gripper code is translated into operation commands to manipulate the fingers. Figure 6.1 shows the conceptual relations required to achieve the goal.

The information transformation from the gripper code to a series of operation commands is made through the interpreter which comprises a number of IF-THEN rules defining the transformation. Let us still take the example of the object-1 in Figure 3.9. Its matrix is shown in Section 5.2. According to the matrix, the gripper code is generated as

\[
\begin{bmatrix}
2 & 2 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}
\]

Where, the non-zero digits appear in the first layer and first two columns indicating that only the two primary fingers are going to operate and in the two fingers only their joints A to be actuated in gripping. That means to grasp a small size block like the object-1,
only the linear translating motion of the joint A is needed and no rotating and yawing motions are required. According to the code digits, the following operation commands are generated through the interpreter.

1. The joint A of the finger 1 translates about 20 mm, and
2. The joint A of the finger 2 also translates about 20 mm from their origins respectively.
3. The joint B and C of these two fingers do not move, and
4. The finger 3 does not move. In practice it moves away from the centre of gripping in order to avoid interference.

The origin of the fingers of the gripper is defined as the close state of its fingers.

6.3 Geometrical Transformation

In order to carry out the grasping task, the gripper will be manipulated from its origin to the working place where the object is located. This can be done by a geometrical transformation which usually consists of a rotation matrix and a translation vector. According to the establishment of correspondence discussed in Chapter 5, the gripper may be transformed into the position and orientation of the object in the scene by means of the rotation matrix and the translation vector [121].
Shown in Figure 6.2, there are two coordinate systems, the relatively fixed coordinate frame OXYZ as a reference frame attached to the centre of the object, the moving (translating and rotating) coordinate frame O'UVW through the tips centre of the gripper fingers. Suppose there is a position vector \( P \) which can be represented by its coordinates with respect to the O'UVW and OXYZ coordinate systems, respectively, as

\[
\begin{align*}
\overset{o}{P} &= (P_u, P_v, P_w) \\
\overset{\theta}{P} &= (P_x, P_y, P_z)
\end{align*}
\]  

(6.1)

The \( \overset{o}{P} \) is translated to \( \overset{\theta}{P} \) by \( t \) as

\[
\overset{\theta}{P} = \overset{o}{P} + t
\]

(6.2)

Where \( t \) denotes the translation coordinates. If the O'UVW coordinate system is rotated an \( \alpha \) angle about the OX axis, \( \varphi \) angle about OY axis and \( \vartheta \) angle about OZ axis with respect to the OXYZ coordinate system, the 3x3 rotation matrices are, respectively,
\[
R_{x,\alpha} = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos\alpha & -\sin\alpha \\
0 & \sin\alpha & \cos\alpha
\end{bmatrix}
\quad
R_{y,\varphi} = \begin{bmatrix}
\cos\varphi & 0 & \sin\varphi \\
0 & 1 & 0 \\
-\sin\varphi & 0 & \cos\varphi
\end{bmatrix}
\quad
R_{z,\theta} = \begin{bmatrix}
\cos\theta & -\sin\theta & 0 \\
\sin\theta & \cos\theta & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

(6.3)

So the point \( P \) in the \( O'U'V'W' \) coordinate system is transformed to the \( OXYZ \) coordinate system by the 3x3 rotation matrices \( R_{x,\alpha}, R_{y,\varphi}, R_{z,\theta} \) and a translation vector \( t \) as

\[
{o}P = R{o}P + t
\]

(6.4)

where, \( R = R_{x,\alpha}R_{y,\varphi}R_{z,\theta} \).

The above equation can be further expressed by the homogeneous transformation matrices [121].

6.4 Gripper Control

The operation commands for actuating the three joints of a finger are carried out through three DC motors respectively, which are controlled by a computer. The control schematic of one finger is shown in Figure 6.3. The commands are coded by C++ program.
The control is taken in three layers: joint A, joint B and joint C respectively. Based on the control, the rough grasp is established with respect to the gripper configuration. It is evident that the gripper configuration is directly correspondent to a relevant gripping model - object gripping family, and the rough grasp is a detailed representation of the configuration with respect to a particular object in the gripping family. The fine grasp is not considered in this study.

The control circuit for driving a DC motor is given in Figure 6.4. The circuit can control the turning of the motor in two directions through a transistor centred circuit (terminal A), and start and stop the motor through a relay (terminal B). The digital signals through the terminal A and B are provided by the control computer.
The experiment setup for the gripper control is shown in Figure 6.5.

![Figure 6.5 Experiment setup of the gripper control](image)

6.5 Conclusion

The gripper code presented in the chapter is an extraction of the gripper configuration matrix. It does not repeat the information represented in the matrix but further details the gripping information. The code concentrates on the information related to the gripper joints and their motions, which are directly connected with the gripper operation and control. The code is as a means to convey the gripping features into the joints operations. That is, different face types require different joints to be operated.
The operation commands generated in terms of the gripper code are used for the computer to control the gripper performance. The control is to establish a rough grasp. The fine grasp will be done when the fingers actually touch the object faces with a sensory feedback system implemented with the gripper. The three layers of control for joint A, B and C respectively make the control strategy clear and easily implemented. The control is divided into two steps. One step is to move the gripper to the scene and the other one is to move its fingers to their desired positions for establishment of the rough grasp. Manipulation of the gripper is made by means of DC motors which actuate each joint of the gripper. The joint motion here is only taken into account its kinematics, not dynamics.
CHAPTER 7  GRIPPER DESIGN

7.1 Introduction

Gripper design is involved with handling and assembly tasks, object shape and geometry, kinematic and dynamic motions, joint link structures and fingertip types, as well as gripping control strategy. As mentioned before, the grasping strategy proposed in the system is object oriented. According to the object oriented concept, it is desired that a simple grasp can be utilised for a wide variety of objects and the gripper would be designed flexible and as simple as possible in its structure. In order to achieve this goal, the following considerations are taken into the gripper design.

- Grasp configuration can be suitable for a large number of objects with different shapes.
- Each joint of a finger has only one type of motion.
- The driven-chain mechanism is simple and easy to be implemented and maintained.
- The control scheme is simple.
- The cost of the gripper design is low.

7.2 Gripper Structure

The gripper developed is composed of three fingers and each finger has three joints with same structure. Each joint is restricted to only one type of motion. The gripper is shown in Figure 7.1 and its one finger structure in Figure 7.2. The element design of the gripper is given in Appendix D.
Figure 7.1 The gripper with three fingers

Figure 7.2 A finger structure detailed design
The finger can be divided into three component parts corresponding to the three joints. The first part is composed of a DC motor, a speed reducing gear system, a leadscrew and a moving threaded block, and other supporting elements. The speed reducing gear system includes a planetary gear box and a pair of output external gears. In addition to the speed reducing, the external gears can change the positions of the motor shaft and the output shaft so that a potentiometer is easy to be mounted for position control. As power is provided by the motor, enough force is generated through the gearing system to rotate the leadscrew and then a translating movement is produced by the motion transmission through the threaded block. The friction is high between the pair of the screw shaft and the threaded block because nearly half of the finger weight is supported by the pair. Therefore, the materials of the pair is required to be of low friction effects. Here the aluminum and plastic are employed.

The second part is only composed of a DC motor, a planetary gear box and a pair of external gears. The joint B can be rotated in 360 degree in two directions. This makes the fingers suitable for those objects with round like surfaces or polyhedrons. The third part consists of a DC motor, a planetary gear box, a leadcrew and a finger rod. The finger rod has a tip directly to touch the grasped face of an object. The rotating motion of the gear output shaft can be transmitted to the yawing motion of the finger rod through the leadscrew. The yawing range is designed about 0 - 30 degree. The detailed dimensions of a finger is referred to Appendix C. The control of the finger is already discussed in the Section 6.4. Here it should be emphasized that each joint is actuated by one motor respectively. One joint motion is independent of other two joints. How many distance it should move or how many degrees it rotate is fully determined by the gripping model and the face types. So the control of each joint is relatively simple and straight.

The three fingers are arranged in the positions forming a triangle shown in Figure 7.3.
The finger 1 and finger 2 are defined as the primary fingers and the finger 3 as an auxiliary one. For any grasping, the first consideration is to try to use the two primary fingers and then the auxiliary finger.

7.3 Analysis of Grasping Force

The grasping force of the gripper is generated by electric actuators - DC motors through a transmission mechanism. Here the relationship between the actuating force (input) and the gripping force (output) is simply discussed. The force analysis is made with respect to three joints respectively by using the classical static force analytical method.

For the joint A, its force transmission mechanism may be simply depicted in Figure 7.4.
In Figure 7.4, $T$ is the input torque of the joint A which is obtained from the gearing output of the part. $F_g'$ denotes the grasping force generated by $T$ through the leadscrew drive. $Q_T$ denotes the push force acting on the threaded block to produce the grasping force. According to the screw force transmission relation, $Q_T$ may be expressed as [122] [123]

$$Q_T = \frac{2\pi \eta}{l} T$$  \hspace{1cm} (7.1)$$

where $l$ is the lead of threads, and $\eta$ is the screw efficiency. Applying the force equilibrium in the Figure 7.4 (b), one can obtain the grasping force as

$$F_g' = Q_T = \frac{2\pi \eta}{l} T$$  \hspace{1cm} (7.2)$$
The joint B transmits the rotating motion by gear drives. The external gear and the planetary gear drives are depicted in Figure 7.5.

Assume that the teeth numbers of the gear 1 and gear 2 in Figure 7.5 (a) are $N_1$ and $N_2$, and the angular velocities of the gears are $\omega_1$ and $\omega_2$ respectively. Since torque is proportional to the radius of the gear, the following equation can be derived [123] [124].

$$\frac{T_1}{T_2} = \frac{N_1}{N_2} = \frac{\omega_2}{\omega_1} \quad (7.3)$$

where $T_1$ and $T_2$ are torques acting on the gear 1 and gear 2 respectively. For the planetary gear trains in Figure 7.5 (b), assume that the teeth numbers of the gear 1 and gear 2 are $N_1$ and $N_2$, and the angular velocities of the gear 2 and the arm 4 are $\omega_2$ and $\omega_4$ respectively. The ring gear 1 is stationary. According to the planetary gearing [122] [125], the relationship of the torques $T_2$, $T_4$ and the velocity ratio can be given as
\[ \frac{T_4}{T_2} = \frac{\omega_4}{\omega_4} = 1 + \frac{N_1}{N_2} \quad (7.4) \]

Actually, the rotating force needed for the joint B is very small.

The force diagram of the joint C may be illustrated as Figure 7.6.

![Figure 7.6 Force diagram of joint C](image)

In Figure 7.5 the push force \( Q_T \) in the screw is generated by actuating torque \( T \) to drive the threaded block down in order that the finger rod has the grasping force \( F_G'' \) enough to perform the grasping operations. Applying the torque equilibrium to Figure 7.5 (b) at point A, one can have the equation

\[ F_G'' \cos \alpha \cdot L - Q_T \cdot l = 0 \quad (7.5) \]
Thus the grasping force is given as

\[ F_G = \frac{1}{L \cos \alpha} Q_T \]  \hspace{1cm} (7.6)

where the angle \( \alpha \) is the yawing angle of the finger rod.

In terms of the Eq. (7.2) and (7.6), the grasping force \( F_G \) may be expressed in a vector form as

\[ \overrightarrow{F_G} = \overrightarrow{F'_G} + \overrightarrow{F''_G} \]  \hspace{1cm} (7.7)

7.4 Conclusion

The gripper design is mainly concerned with its mechanical structure and control mechanism. The mechanical structure is to provide mechanical links and transmission for the gripper so that the fingers of the gripper have sufficient degrees of freedom in motion, and ensure that their configurations can suit a variety of objects with different surface characteristics and shapes. The control mechanism is to provide a simple and efficient manipulation to drive the fingers of the gripper to their desired grasping positions and to perform the grasping tasks.

The gripper is designed with three fingers and each finger has three joints. Each joint has one type of motion, i.e. one degree of freedom. Thus the gripper has totally nine degrees of freedom. According to the grasping tasks and geometrical properties, all objects to be grasped are classified into six gripping models of a block, polyhedron, cylinder, sphere, cone and wedge in the system. The flexibility of the gripper designed makes the finger configurations suitable for the six different models. Therefore, it is possible for the gripper to carry out the grasping tasks for a large number of objects with different shapes.
The force transmission mechanism designed for the gripper is quite simple. It is composed of DC motors, gear, screw trains, and simple linkages which are easily purchased or manufactured. Since each joint is actuated by a DC motor separately, the kinematic control of the gripper is easy to be achieved.
CHAPTER 8 CONCLUSION

8.1 Introduction

This final chapter is to briefly summarise and evaluate the entire studies. The contributions of the thesis and the significance of the studies are discussed and evaluated. The program structure for the integrated system from image collection, part recognition to code generation and grasping analysis is introduced. The future work in this area is proposed.

8.2 General Discussion of the Study

Grasping analysis and operation are indeed quite complicated. They are concerned with many aspects such as the task requirements in gripping for part handling or assembly, the geometry of objects to be grasped, the approaches to recognise objects in the scene including their positions and orientations, the methods to represent the objects and grasping and to transfer information between the objects and a gripper for performing the grasping tasks, and the gripper design and control.

This study, using a systematic way, integrates all aspects above as a whole. The system developed is around the centre of grasping. The grasping features connect the image data, object models, gripping models, coding system and the gripper configurations and operations, and guide the information flow of the system. The structure of this system is a hierarchy consisting of several levels of abstraction, i.e. object recognition and model description - grasping feature representation and classification - matching strategy for the objects and the gripper - grasping description and operation. The analysis of the system is object-oriented, i.e. grasping-oriented.

There are two models represented in the system: object models and gripping models. The object models are used for object recognition with respect to the image data from the
scene. The image features are extracted based on the object shapes represented by means of moment invariants which provide not only the shape features of the object but also the position and orientation of the object. In addition, the object models also provide the geometrical information for establishment of the gripping models through the face relation graphs which particularly concern the grasping face relations. Actually, the gripping models are extracted from the relevant object models and they classify objects to be grasped into different gripping families in terms of the geometrical and grasping properties of the objects. Based on the gripping models a large number of objects may be classified and grouped into a few families according to their geometrical and grasping similarities. This would increase the efficiency of grasping analyses and operations.

According to group technology, the object gripping models and features are represented by coding digits in the coding system proposed. The coding system does not describe the object geometry in detail as other coding system do, but catches the object features related to gripping. The grasping features include the primary features of the object (corresponding to its gripping model), the faces types to be touched by the gripper fingers and the gripping attributes. The coding digits are generated by knowledge representation. The relations between the grasping features and the corresponding digits are represented by IF-THEN rules. The inference to find relevant digits starts from the searching of the gripping model after the object is recognised in the scene.

The establishment of correspondence and the transformation of gripping information between objects and the gripper are accomplished through a matrix description and knowledge reasoning. The object and gripper matrices are proposed as a means to connect the gripper and the object to be grasped. The object matrix describes the object grasping features with respect to the faces to be touched by the gripper fingers. It details the grasping information provided by the coding digits. The description comprises the number of faces selected for gripping, the faces types, the faces distance and the touch points of the faces. Correspondingly, the gripper matrix represents the fingers positions and the gripper configuration in gripping. It indicates the number of fingers required for
grasping, the fingers configurations, the fingers positions and the tips positions. The matrix representation provides a way to describe the gripper, the object and their relation explicitly, and establish correspondence between them efficiently. Based on the matrices, the transformation of the grasping information from the object to the gripper, i.e. the matching of the two matrices is made through knowledge reasoning.

In light of the gripper matrix, a gripper code is produced for detailed description of the joint motions of the gripper. And then a series of operation commands are generated for manipulation of the gripper with respect to the gripper code.

The specific gripper with three fingers is developed for demonstration of the concept proposed in this study. The proposed system is tested successfully. Obviously, the generic concept and the integrated system structure could also be applicable to other types of grippers, even dextrous grippers.

8.3 Program Implementation

The program is written by Visual Basic 4.0 which is an object-based, event-driven programming language. The program provides a user-friendly interface, is easily run and controlled by users step by step. It comprises four forms, namely Image Data Collection, Object Recognition, Matrix Generation and Gripper Command. The four forms actually indicate the four major parts of the system.

In the form of Image Data Collection, there are four controls: PC Board Setup, Open File, Read and Collect Data, Data Thresholding, and Data Smoothing and Edge Detecting. The PC Board setup is to initialise the PC-30B input and output board and set necessary parameters for the board. The rest three controls are used for the events of image data collection and processing.
In the form of Object Recognition, there are controls of Feature Extraction, Object Recognition and Object Code Generation. In these events, several important procedures are made. Object features are extracted from the image; the object in the scene is recognised with respect to object models; the object code is generated with respect to the object gripping features; and the object is classified into a specific grasping family. Knowledge-based inference is applied to the object recognition and code generation. In the inference, the process of an IF-THEN reasoning is directly represented by the Visual Basic code.

In the form of Matrix Generation, the object and gripper matrices are generated and displayed in the screen in terms of the information from the object gripping model and its coding digits. The information and procedures transferred between forms are represented by global statements provided by Visual Basic.

The last form of Gripper Command includes controls of Gripper Code Generation and Gripper Operations. The control of the gripper operations is written by Borland C++ 3.1 program which, as an EXE file, is embedded in the Visual Basic program. C++ has advantage of ease communication with computer I/O ports.

8.4 Proposed Future Work

Although many work has been done in the studies, there are still some aspects of the research needed to be further investigated because the time available for the research is limited. On the other hand, research solves problems but new questions are generated again in research. Future work on image processing, grasping representation and knowledge inference, and the gripper modification is proposed as follows.

1. At present the image processing and analysis are two-dimensional. As far as complicated objects are concerned, it is better to analyse the image data in three-dimensions, or use more than one image data from different sides of view of an object to
do two- or three-dimensional analysis simultaneously and coordinately. Thus the grasping features of the object may be directly extracted from the image analysis, or those grasping features from the image can be as suggestions for further decision making.

2. Grasping representation plays a key role in the knowledge-based system of this study. The integration of coding digits and production rules is a new way to represent and reason the grasping knowledge in the system developed. The future work in the aspect may involve to develop an expert system based on the current knowledge-based analysis and reasoning, and to investigate a unified representation for object geometry, grasping features and grasping operations. Object oriented analysis and programming may be a way to go in this regard [118].

3. For the gripper designed and developed in the study, it is necessary to continue to refine it in structure and control, especially in its sensory system so that a fine grasping can be achieved properly. The dynamic analysis and control of the gripper are also needed to be studied. Furthermore, it is necessary to do more experimental testing in order to apply the concept and the approaches proposed in the studies to actual manufacturing sites.
APPENDIX A

In the object matrix shown in Figure 5.1, the detailed values of $a_{2j}$, $a_{3j}$ and $a_{4j}$ are given as follows.

$a_{2j}$ indicates a face type. There are following face types and corresponding $a_{2j}$ values.

<table>
<thead>
<tr>
<th>Face type</th>
<th>$a_{2j}$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planar face</td>
<td>1</td>
</tr>
<tr>
<td>Wedged face</td>
<td>2</td>
</tr>
<tr>
<td>Cylindrical face</td>
<td>3</td>
</tr>
<tr>
<td>Spherical face</td>
<td>4</td>
</tr>
<tr>
<td>Conical face</td>
<td>5</td>
</tr>
<tr>
<td>Free curved face</td>
<td>6</td>
</tr>
</tbody>
</table>

$a_{3j}$ indicates the distance between faces and the centre of the object. For instance, for a two parallel faces the distance is measured as the half distance of the two faces; for a cylinder, the distance is its radius. The actual distance is divided into five ranges, namely a very small, small, medium, large and very large one. The corresponding $a_{3j}$ values are given as

<table>
<thead>
<tr>
<th>Distance range (mm)</th>
<th>$a_{3j}$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very small:</td>
<td>&lt;10 1</td>
</tr>
<tr>
<td>Small:</td>
<td>11 - 20 2</td>
</tr>
<tr>
<td>Medium:</td>
<td>21 - 30 3</td>
</tr>
<tr>
<td>Large:</td>
<td>31 - 40 4</td>
</tr>
<tr>
<td>Very large:</td>
<td>&gt;41 5</td>
</tr>
</tbody>
</table>

$a_{4j}$ indicates the contact point of a face with the tip of a finger. The point is measured in terms of the distance between the point and the face edge where the gripper is supposed to approach in, refer to Figure A.1. Depending on its accuracy, the distance is also divided into five ranges as follows.
<table>
<thead>
<tr>
<th>Distance range (mm)</th>
<th>$a_{ij}$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very small: &lt;10</td>
<td>1</td>
</tr>
<tr>
<td>Small: 11 - 20</td>
<td>2</td>
</tr>
<tr>
<td>Medium: 21 - 30</td>
<td>3</td>
</tr>
<tr>
<td>Large: 31 - 40</td>
<td>4</td>
</tr>
<tr>
<td>Very large: &gt;41</td>
<td>5</td>
</tr>
</tbody>
</table>

Figure A.1 The distance $a_{ij}$ measurement
APPENDIX B

In the gripper matrix shown in Figure 5.3, the detailed values of $b_{2j}$, $b_{3j}$ and $b_{4j}$ are given as follows.

$b_{2j}$ indicates a finger configuration. It has a close relation with a face type on which the finger should touch in gripping. There are following face types and corresponding predefined $b_{2j}$ values.

<table>
<thead>
<tr>
<th>Face type</th>
<th>$b_{2j}$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planar face</td>
<td>1</td>
</tr>
<tr>
<td>Wedged face</td>
<td>2</td>
</tr>
<tr>
<td>Cylindrical face</td>
<td>3</td>
</tr>
<tr>
<td>Spherical face</td>
<td>4</td>
</tr>
<tr>
<td>Conical face</td>
<td>5</td>
</tr>
<tr>
<td>Free curved face</td>
<td>6</td>
</tr>
</tbody>
</table>

$b_{3j}$ indicates the finger position relative to the centre of the gripper when a grasping operation is prepared to be performed. The finger position is measured in terms of the distance between a finger (here means its joint A) and the centre of the gripper. The distance is also divided into five ranges, namely a very small, small, medium, large and very large, with respect to its relevant object matrix. The $b_{3j}$ values are given as

<table>
<thead>
<tr>
<th>Distance range (mm)</th>
<th>$b_{3j}$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very small:</td>
<td>&lt;10</td>
</tr>
<tr>
<td>Small:</td>
<td>11 - 20</td>
</tr>
<tr>
<td>Medium:</td>
<td>21 - 30</td>
</tr>
<tr>
<td>Large:</td>
<td>31 - 40</td>
</tr>
<tr>
<td>Very large:</td>
<td>&gt;41</td>
</tr>
</tbody>
</table>

$b_{4j}$ indicates the tip position of a finger, which is in correspondence with the contact point of a face to be grasped. The tip position is measured in terms of the distance between the contact point of the face and the face edge where the finger tip starts.
approaching into the face, refer to Figure A.1. The distance ranges and their corresponding $b_{4j}$ values are as follows

<table>
<thead>
<tr>
<th>Distance range (mm)</th>
<th>$b_{4j}$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very small:</td>
<td>&lt;10</td>
</tr>
<tr>
<td>Small:</td>
<td>11 - 20</td>
</tr>
<tr>
<td>Medium:</td>
<td>21 - 30</td>
</tr>
<tr>
<td>Large:</td>
<td>31 - 40</td>
</tr>
<tr>
<td>Very large:</td>
<td>&gt;41</td>
</tr>
</tbody>
</table>


APPENDIX C

The gripper code shown in Table 6.1 describes the joint motions of the gripper. The translating of the joint A is coded as

<table>
<thead>
<tr>
<th>Digit</th>
<th>Translating (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
</tr>
</tbody>
</table>

The rotating of the joint B is coded as

<table>
<thead>
<tr>
<th>Digit</th>
<th>Rotating (degree)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>45</td>
</tr>
<tr>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td>5</td>
<td>75</td>
</tr>
</tbody>
</table>

The yawing of the joint C is coded as

<table>
<thead>
<tr>
<th>Digit</th>
<th>Yawing (degree)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
</tr>
</tbody>
</table>
APPENDIX D

The element design of the gripper is given as follows.

Element No. 1
Material: Aluminium
Element No. 2
Material: Aluminium

Element No. 3
Material: Plastic
Element No. 5
Material: Aluminium

Element No. 6
Material: Aluminium
Element No. 8
Material: Aluminium

Element No. 9
Material: Aluminium
Element No. 11
Material: Aluminium

Element No. 12
Material: Aluminium
Element No. 13
Material: Aluminium

Element No. 14
Material: Aluminium
Element No. 15
Material: Aluminium
REFERENCES


Assembly and Machining in Polyhedral Representation, International Journal of

[15] Ambler, A. P., and Popplestone, R. J., (1975), inferring the Positions of
Bodies From Specified Spatial Relationships, Artificial Intelligence, No. 6,


for a Language for Describing Assemblies, Artificial Intelligence, No.14,
pp. 79 - 109.

Representing Assemblies: Part 1, Computer-Aided Design, Vol. 17,
No. 1, pp. 15 - 19.

[19] Lee, K., and Gossard, D. C., (1985), Inference of the Positions of
Components in an Assembly: Part 2, Computer-Aided Design, Vol. 17, No.1
pp. 20 - 23.


Mechanical Assembly Sequences, IEEE Journal of Robotics and

E., (1991), An Integrated Computer Aid for Generating and Evaluating
Assembly Sequences for Mechanical Products, IEEE Transactions on


[34] Homem de Mello, L. S., and Sanderson, A. C., (1990), Evaluation and
Selection of Assembly Plans, Proceedings IEEE Conference on Robotics
and Automation, pp. 1588 - 1593.

Mechanical Assembly Sequences, IEEE Transactions on Robotics and

Application of Theorem Proving to Problem Solving, Artificial Intelligence, No.2,
pp. 189 - 208.

Problem Solving, Machine Intelligence, Vol. 3, pp. 405 - 430.

[38] Fikes, R. E., Hart, P. E., and Nilsson, N. J., (1972), Learning and Executing

Planning, Tutorial on Robotics, Lee, C. S. G., Gonzalez, R. C., and Fu, K. S.,

Intelligence, Vol. 5, pp. 1 - 49.


System for Technological Planning in Robot-Based Flexible Assembly Systems,


of the CIRP, Vol.41/2, pp. 625 - 636.


Press Ltd., Hampshire and London.


