

University of Warwick institutional repository: <http://go.warwick.ac.uk/wrap>

**A Thesis Submitted for the Degree of PhD at the University of Warwick**

<http://go.warwick.ac.uk/wrap/3674>

This thesis is made available online and is protected by original copyright.

Please scroll down to view the document itself.

Please refer to the repository record for this item for information to help you to cite it. Our policy information is available from the repository home page.

**A FEATURE-BASED  
REVERSE ENGINEERING SYSTEM USING  
ARTIFICIAL NEURAL NETWORKS**

*A dissertation submitted by*

**YONG-TAE JUN**

*Submitted in fulfilment of the requirements  
for the Degree of Doctor of Philosophy*



**SCHOOL OF ENGINEERING  
UNIVERSITY OF WARWICK**

June 1999

To the major candles in my life:

*my wife Sung-Hee,*

*my son Jae-Joon, and*

*my daughter Hye-Jin*

# Table of Contents

<b>List of Figures</b>	vii
<b>List of Tables</b>	x
<b>Acknowledgement</b>	xi
<b>Declaration</b>	xii
<b>List of Publications</b>	xiii
<b>Summary</b>	xv
<b>Acronyms</b>	xvi
<b>Glossary</b>	xviii
<b>Chapter 1 Introduction</b>	<b>1</b>
1.1    Definitions of Reverse Engineering	2
1.2    Statement of Problems	3
1.3    Aims of the Research	5
1.4    Research Methodology	6
1.4.1    Data acquisition and reduction	7
1.4.2    Edge detection	7
1.4.3    Segmentation	8
1.4.4    Free form surface creation	8
1.4.5    Feature recognition	9
1.4.6    Feature extraction	10
1.4.7    CAD model exchange	10
1.5    Assumption of the Research	11
1.6    Thesis Outline	12
<b>Chapter 2 Review of Reverse Engineering Technology</b>	<b>13</b>
2.1    Why Use Reverse Engineering	14
2.2    The Scope of Reverse Engineering	17
2.2.1    Surface generation	18
2.2.2    Tool-path generation	20



2.2.3	Feature analysis and inspection	21
2.3	Data Acquisition	23
2.4	Scanning Devices	25
2.4.1	Contact scanning devices	25
2.4.2	Non-contact scanning devices	28
2.4.3	Comparisons of contact and non-contact scanning method	31
2.5	Current Issues associated with Reverse Engineering	34
2.5.1	Processing a large amount of scanned data	34
2.5.2	Automatic extraction of 3D CAD models from 2D data	34
2.5.3	Automatic extraction of 3D CAD models from 3D data	35
2.5.4	Edge detection	37
2.5.5	Noise and missing data	37
2.5.6	Data exchange	38
2.5.7	User Interface	39
 <b>Chapter 3 Scanned Data Processing and Edge Detection</b>		<b>40</b>
3.1	Data Structure for Point Clouds	44
3.1.1	POINT CLOUD data structure	48
3.1.2	MESH structure	49
3.1.3	STRIP structure	50
3.1.4	POINT structure	51
3.2	Filtering Algorithms	52
3.3	Edge Detection	56
3.3.1	Sharp edge detection by vector manipulations	58
3.3.2	Smooth edge detection	60
3.4	Sampling Algorithms	67
3.4.1	Sampling by vector manipulations	68
3.4.2	Sampling by rate	72
3.4.3	Sampling by voxels	73
3.5	Data Segmentation	74

<b>Chapter 4 Surface fitting to Scanned data using B-spline techniques</b>	<b>75</b>
4.1 The Basic of Surface Modelling	76
4.1.1 The characteristics of surface modelling	76
4.1.2 Surface interpolation and approximation	77
4.1.3 Surface modelling techniques	78
4.2 Surface Fitting Algorithms in this Research	79
4.3 Fitting a B-spline Surface to a set of data points	81
4.3.1 The definition of B-spline basis functions	81
4.3.2 Parameterisation	83
4.3.3 Selection of knots	85
4.3.4 Insertion of points	87
4.3.5 Surface optimisation	88
4.3.6 Advantages of B-spline representation	90
4.4 B-spline Data Structure	91
4.4.1 The data structure of B-spline curve	92
4.4.2 The data structure of B-spline surface	92
4.5 Trimmed Surfaces	93
4.6 B-spline Entities within Standard File Formats	97
4.7 B-spline Surface Fitting Examples	98
<b>Chapter 5 Boundary-Representations for Reconstructed Parts</b>	<b>102</b>
5.1 The Basic of B-rep based modelling	104
5.2 The Data Structure of B-rep	106
5.2.1 Euler–Poincare’ law	106
5.2.2 Manifold and non-manifold	107
5.2.3 B-rep model representation	108
5.2.4 Topological data structure	110

<b>Chapter 6 Feature-based Technology in Reverse Engineering</b>	<b>113</b>
6.1 Feature-based Technology	115
6.2 Definition of Feature	116
6.3 Advantages of Feature-based Approach	118
6.4 Feature Relationships	119
6.4.1 Parameter inheritance	119
6.4.2 Adjacency relationships	119
6.4.3 Placing of features	119
6.5 Feature Database in the Application	120
6.6 Feature Library	122
6.7 Feature Recognition Methods	127
6.7.1 Graph-based recognition	128
6.7.2 Set-based recognition	129
6.7.3 Ruled-based recognition	129
6.7.4 Syntactic-based recognition	130
6.8 The Proposed Feature Recognition Method	131
<b>Chapter 7 Feature Recognition using Artificial Neural Networks</b>	<b>133</b>
7.1 Introduction to ANNs	137
7.1.1 A simple neuron	137
7.1.2 Training by Back-Propagation algorithms	141
7.1.3 What are the differences between ANNs and conventional procedural computing?	142
7.2 Previous Work in Feature Recognition using ANN	144
7.3 The Proposed Feature Recognition Procedure	145
7.4 The ANN Module in this Research	148
7.4.1 The ANN structure	148
7.4.2 Training	150
7.5 Deriving Geometric Attributes for Feature Recognition	154
7.5.1 Chain code attribute	156
7.5.2 Convex/concave attribute	160

7.5.3	Circular/rectangular attribute	165
7.5.4	Open/closed attribute	170
<b>Chapter 8</b>	<b>CAD Model Reconstruction and Data Communication</b>	<b>172</b>
8.1	CAD Model Reconstruction	173
8.1.1	Maximum-likelihood estimation (MLE) method	174
8.1.2	Parametric design	179
8.2	Data Communication	182
8.2.1	An outline of the IGES standard	183
8.2.2	An outline of the STEP standard	187
<b>Chapter 9</b>	<b>System Implementation and Examples</b>	<b>192</b>
9.1	System Implementation	192
9.2	Example	196
9.2.1	Point data capture	197
9.2.2	Point data processing and edge detection	199
9.2.3	Feature recognition	200
9.2.4	Feature extraction	203
<b>Chapter 10</b>	<b>Discussion and Conclusions</b>	<b>211</b>
10.1	General Discussion	211
10.2	Detailed Discussion and Conclusions According to Sub-Modules	214
10.3	Limitation of the Proposed Application	224
10.4	Further Research	225
<b>References</b>		<b>226</b>
<b>Bibliography</b>		<b>238</b>



## Appendices

Appendix 1	Digitisers	240
Appendix 2	Commercial RE Packages and Service Providers	244
	(a) Commercial Reverse Engineering Packages	244
	(b) Service Providers	249
	(c) Companies dealing with RE Technology	250
Appendix 3	Mathematical Representation for Analytic Entities	255
	(a) Parametric representation for analytic entities	255
	(b) B-spline representation for analytic entities	257
Appendix 4	Back-Propagation (BP) Algorithms	258
	(a) Training parameters	258
	(b) BP training algorithm	260
Appendix 5	Mathematical Algorithms of MLE for Plane and Cylinder	266
	(a) Maximum-likelihood fitting a plane	266
	(b) Maximum-likelihood fitting a cylinder	269
Appendix 6	IGES File Overview	270

## List of Figures

Figure 1.1	Research methodology in the ReTool system	6
Figure 2.1	Scanning hands to making goal-keeping gloves	16
Figure 2.2	The scope of reverse engineering	17
Figure 2.3	Tool path generation from scanned data	21
Figure 2.4	Several configurations of CMMs	26
Figure 2.5	Contact scanning device	27
Figure 2.6	Spot ranging scanning device	28
Figure 2.7	Comparison of contact versus non-contact device at vertical faces	32
Figure 3.1	The data processing steps	41
Figure 3.2	Point cloud data structure	45
Figure 3.3	Three possible scanning strategies for a test part	46
Figure 3.4	Typical characteristics of scanned points and curvatures	54
Figure 3.5	Types of edge points in <i>reverse engineering</i>	57
Figure 3.6	Sharp edge detection process ( $\Phi$ is an angle criterion)	59
Figure 3.7	Circle through three given points	62
Figure 3.8	Point data reduction by vector calculation	68
Figure 3.9	Sampling dialogue box	70
Figure 3.10	Sampling procedure	71
Figure 3.11	Sampling by sampling rate ( <i>sampling rate = 2 and # of iter. = 2</i> )	72
Figure 4.1	<i>B-spline</i> surface fitting flowchart	79
Figure 4.2	Topological and geometric elements in <i>B-rep based</i> solid models	91
Figure 4.3	A surface with a trimmed surface	93
Figure 4.4	Curves on a surface (trimming curves)	94
Figure 4.5	Trimming curves keeping the face on the left	96
Figure 4.6	1290 input data points	98
Figure 4.7	Fitting a <i>NURBS</i> surface to 1290 data points	99
Figure 4.8	Shaded image of reconstructed <i>NURBS</i> surface	100
Figure 4.9	Defining a four corners and creating a surface	100
Figure 5.1	Basic constituents of <i>B-rep</i> models	105
Figure 5.2	Two non-manifold objects	107

Figure 5.3	Winged-edge structure	108
Figure 5.4	Half-edge structure	109
Figure 6.1	An example of feature hierarchy	121
Figure 6.2	A solid model with <i>face adjacency graph</i>	128
Figure 6.3	Syntactic based feature recognition	130
Figure 6.4	The comparison of two feature extraction methods	132
Figure 7.1	Flow chart of proposed feature recognition process	135
Figure 7.2	Diagram of a three-layer neural network	137
Figure 7.3	A <i>unipolar sigmoid</i> function used as a non-linear activation ftn	140
Figure 7.4	The output shell environment for recognised features	146
Figure 7.5	The output shell environment for extracted features	147
Figure 7.6	Neural Network user interface	148
Figure 7.7	A user interface to generate training conditions	150
Figure 7.8	The learning rate	151
Figure 7.9	Implemented neural network structure	155
Figure 7.10	Chain codes for a wedge	156
Figure 7.11	Example chain codes for target generic features	157
Figure 7.12	Chain codes representation along to scanned points	158
Figure 7.13	Chain code normalisation	160
Figure 7.14	Vector manipulations of the convex/concave attribute at a scan line	161
Figure 7.15	Flowchart of the determination of the convex/concave attribute	163
Figure 7.16	The shape attribute examples	165
Figure 7.17	A set of scanned points, the current arc centre and the radius $R$	166
Figure 7.18	The convergence rate for a tested circle	169
Figure 7.19	Curves illustrating the open/closed cases	170
Figure 8.1	Sample parametric definition	180
Figure 8.2	Dialogue box of parametric design for a boss feature	181
Figure 8.3	CAD data exchange using IGES	184
Figure 8.4	Export the test part to commercial CAM system using IGES	185
Figure 8.5	An EXPRESS model within a CAD/CAM environment	188
Figure 8.6	STEP classification of form features	190
Figure 8.7	Volume feature types	191



Figure 9.1	The entire system architecture	193
Figure 9.2	The <i>GUI</i> of the proposed system	194
Figure 9.3	Dialogue box for reading input scanned points	195
Figure 9.4	Example of a part model	196
Figure 9.5	The <i>Cyclone</i> touch trigger	197
Figure 9.6	The cloud of scanned data for the test part	198
Figure 9.7	Data reduction and edge detection	200
Figure 9.8	Boss and hole feature recognition	202
Figure 9.9	Feature extraction	203
Figure 9.10	Feature hierarchy	204
Figure 9.11	Tool path simulation	205
Figure 9.12	NC machining	206
Figure 9.13	The duplicated test part after machining	207
Figure 9.14	Reference points	208
Figure A4.1	Diagram of a three-layer neural network	260
Figure A6.1	A sample ASCII IGES file	271



## List of Tables

Table 2.1	Comparison of contact and non-contact scanning method	33
Table 4.1	Comparisons of B-spline curve definition in IGES and STEP	97
Table 6.1	Generic feature taxonomy in the feature library	124
Table 7.1	A training attribute (one epoch)	152
Table 7.2	Convex/concave attribute	164
Table 9.1	Error analysis results	209
Table A6.1	Format of the DE section in the ASCII IGES form	273
Table A6.2	The PD section of B-spline surfaces	275

## Acknowledgements

I would like to express my gratitude to my supervisor, *Mr. Vinesh Raja*, for his invaluable guidance on my doctoral program. His constant and strong advice has enriched this work.

My sincere thanks also go to *Dr. Jay Bal* for his co-ordinated effort of reading and revising this dissertation. He gave me the most rigorous feedback, particularly when the dissertation was going through the piloting stage.

I am grateful to a number of colleagues who have discussed with me on my research during my time in the University of Warwick: *Mr. Christopher Lawrie, Dr. Hyunwoo Shin, Dr. Shujun Zhang, Mr. Paul Grimley, Dr. Kiah Hion Tang* and *Mr. Winners Ko*. I also acknowledge all the staffs of *Warwick Manufacturing Group*.

I thank the members of *Korean Society* in the University of Warwick, especially *Mr. Sunwoo Lee, Mr. Donghoon Shin, Mr. Seyon Jo, Dr. Changhoon Cha* and *Dr. Siwon Jin*. Their kindness will never be forgotten.


Very special thanks go to *Korea Institute of Science and Technology (KIST)* and to the *British Council* for providing financial support for my academic work. I appreciate all the members of *CAD/CAM Centre* in *KIST*, especially *Dr. Hyungmin Rho, Dr. Sehyung Park, Dr. Myonwoong Park, and Dr. Taesoo Kim*.

I sincerely wish to thank all the members of my family, especially my parents, parents-in-law, eldest brother and younger brother-in-law, for their constant support and encouragement. I am deeply indebted to them.

Finally, I wish to express my heartfelt appreciation to my wife, *Sunghee*, my son, *Jaejoon*, and my daughter, *Hyejin*, for their support, sacrifice, patience and continuous understanding. Without their love, this work would not have been possible. This dissertation is dedicated to them.

## Declaration

I declare that the work presented herein is solely my own work with the exception of quoted information that has been acknowledged through references. None of this work has been previously submitted to another institution for any academic degree.

A handwritten signature in black ink, reading "Yong-Tae Jun". The signature is written in a cursive, flowing style.

*Yong-Tae Jun*

June 1999

## List of Publications

### *(1) Associated Publications & Conference Paper*

**Jun, Y.T., Raja, V.H. and Park, S. (1999)** Geometric feature recognition for Reverse Engineering using Neural Networks, *International Journal of Advanced Manufacturing Technology* (submitted and awaiting referee panel's comments).

**Jun, Y.T. and Raja, V.H. (1999)** Extracting geometric attributes directly from scanned data sets for feature recognition, *International Journal of Computer Integrated Manufacturing* (submitted and awaiting referee panel's comments).

**Jun, Y.T. and Raja, V.H. (1999)** Geometric Feature Recognition using ANNs in Reverse Engineering, *5th International Conference on Manufacturing Technology*, Beijing, China, October.

### *(2) Publications & Conference Papers so far*

**Park, S. and Jun, Y.T. (1996)** Toolpath generation for 5-axis machining of an impeller, *Japan-USA Symposium on Flexible Automation*, pp 729 - 734.

**Kim, T.S. and Jun, Y.T. (1994)** Development of a system for generating STEEL block layout for Trim Die, *Hyundai Motor Company Report*.

**Cho, H.D., Jun, Y.T. and Yang, M.Y. (1993)** Five-axis CNC milling for effective machining of sculptured surfaces, *International Journal of Production Research*, 31(11), pp 2559–2573.



- Park, S., Jun, Y.T., Lee, C.W. and Yang, M.Y. (1993) Determining the cutting conditions for sculptured surface machining, *International Journal of Advanced Manufacturing Technology*, 8, pp 61–70.
- Jun, Y.T. and Park, S. (1993) Generation of trimmed surfaces in a surface modelling system, *Korean Society of Mechanical Engineer*, 17(6), pp 1372–1380.
- Jun, C.S., Jun, Y.T. and Park, S. (1993) Development of a CAM system for 5-axis NC machining of sculptured surfaces, *Korean Society of Precision Engineering*, 10(1), pp 52–61.
- Jun, Y.T. and Park, S. (1993) Toolpath generation of a die cavity defined by sculptured surfaces, *Korean Society of Precision Engineering*, 10(3), pp 161–167.
- Cho, H.D., Jun, Y.T. and Yang, M.Y. (1993) A study on the machining of sculptured surfaces by 5-axis CNC milling (Part I: Cutter axis direction vector and post-processing), *Korean Society of Mechanical Engineer*, 17(8), pp 2001–2011.
- Cho, H.D., Jun, Y.T. and Yang, M.Y. (1993) A study on the machining of sculptured surfaces by 5-axis CNC milling (Part II: The prediction of cusp heights and the determination of tool path interval), *Korean Society of Mechanical Engineer*, 17(8), pp 2012–2022.
- Keum, Y.T., Park, S. and Jun, Y.T. (1992) CAD/CAM/CAE system for Autobody Stamping Dies, *Pacific Conference on Manufacturing, Sakai, Japan, Nov. 3-6*.

## Summary

Reverse Engineering (RE) is the process of reconstructing CAD models from scanned data of a physical part acquired using 3D scanners. RE has attracted a great deal of research interest over the last decade. However, a review of the literature reveals that most research work have focused on creation of free form surfaces from point cloud data. Representing geometry in terms of surface patches is adequate to represent positional information, but can not capture any of the higher level structure of the part. Reconstructing solid models is of importance since the resulting solid models can be directly imported into commercial solid modellers for various manufacturing activities such as process planning, integral property computation, assembly analysis, and other applications.

This research discusses the novel methodology of extracting geometric features directly from a data set of 3D scanned points, which utilises the concepts of artificial neural networks (ANNs). In order to design and develop a generic feature-based RE system for prismatic parts, the following five main tasks were investigated. (1) point data processing algorithms; (2) edge detection strategies; (3) a feature recogniser using ANNs; (4) a feature extraction module; (5) a CAD model exchanger into other CAD/CAM systems via IGES.

A key feature of this research is the incorporation of ANN in feature recognition. The use of ANN approach has enabled the development of a flexible feature-based RE methodology that can be trained to deal with new features. ANNs require parallel input patterns. In this research, four geometric attributes extracted from a point set are input to the ANN module for feature recognition: *chain codes*, *convex/concave*, *circular/rectangular* and *open/closed attribute*. Recognising each feature requires the determination of these attributes. New and robust algorithms are developed for determining these attributes for each of the features.

This feature-based approach currently focuses on solving the feature recognition problem based on 2.5D shapes such as *block*, *pocket*, *step*, *slot*, *hole*, and *boss*, which are common and crucial in mechanical engineering products. This approach is validated using a set of industrial components. The test results show that the strategy for recognising features is reliable.

# Acronyms

<i>2D:</i>	Two Dimensional
<i>3D:</i>	Three Dimensional
<i>AI:</i>	Artificial Intelligence
<i>ANN:</i>	Artificial Neural Network
<i>API:</i>	Application Program Interface
<i>ASCII:</i>	American Standard Code for Information Interchange
<i>BP:</i>	Back-Propagation
<i>B-rep:</i>	Boundary representation
<i>CAE:</i>	Computer Aided Engineering
<i>CAD:</i>	Computer Aided Design
<i>CAM:</i>	Computer Aided Manufacture
<i>CAPP:</i>	Computer Aided Process Planning
<i>CCD:</i>	Charge-Coupled Device
<i>CE:</i>	Concurrent Engineering
<i>CMM:</i>	Coordinate Measuring Machine
<i>CNC:</i>	Computer Numerical Control
<i>CSG:</i>	Constructive Solid Geometry
<i>DXF:</i>	Data eXchange Format
<i>GUI:</i>	Graphic User Interface
<i>IGES:</i>	Initial Graphics Exchange Specifications
<i>ISO:</i>	International Organisation for Standardisation
<i>IT:</i>	Information Technology
<i>LCS:</i>	Local Coordinate System



<i>LMS:</i>	Least Mean Square
<i>MLE:</i>	Maximum-Likelihood Estimation
<i>NC:</i>	Numerical Control
<i>NURBS:</i>	Non-Uniform Rational B-Spline representation
<i>PDES:</i>	Product Data Exchange using STEP
<i>QC:</i>	Quality Control
<i>RE:</i>	Reverse Engineering
<i>RP:</i>	Rapid Prototyping
<i>STEP:</i>	STandard for the Exchange of Product data
<i>WCS:</i>	World Coordinate System



## Glossary

*Artificial Neural Networks (ANNs)* is a biologically inspired approach based upon a network of artificial neurons, similar to those of the human brain. The goal of ANN is to emulate human intelligence. ANNs learn by being trained through a process which changes the weights of the connections between their artificial neurons.

*Attributes* denote the geometric patterns of scanned point data such as chain codes, circular/rectangular, convex/concave and open/closed. They are input to the ANN module for feature recognition.

*Back-Propagation (BP)* is a learning or training procedure based on gradient search with the least-sum-square criterion, called a *steepest-descent technique*. BP is the most widely used learning mechanism for multi-layered perceptrons. The training using *BP* is a non-linear optimisation problem, so an iterative process is usually adopted to solve the problem.

*Concurrent engineering (CE)* means the simultaneous development of products and their manufacturing process, with the goal of removing the delays inherent in traditional sequential approach. In this way, CE promises to cut product-development time, improve quality, and reduce costs.

A *Coordinate Measuring Machine (CMM)* is a 3D scanning device based on three orthogonal axes and employ a contact probe to scan the surface of the part. These axes, designated  $x$ ,  $y$  and  $z$ , embody a spatial coordinate system, similar to that of a NC milling machine. It consists of a mount which can hold different

probes which can move in any of three coordinate directions. CMMs have three movable components along three mutual perpendicular guideways.

*Data capture* is to scan geometric information representing the physical component shapes using 3D scanning systems. The system records the physical entities of a component in the form of numerical data. This data is in the form of a point cloud file, with each point representing a  $x$ ,  $y$  and  $z$  position.

*Data translation* is to convert data from one format to another such as IGES, STEP, vendor- and user-supplied data format.

*Edges* in RE are either the outer boundary that constitutes regions in a part with relatively distinct discontinuities, or internal boundaries in a region.

*Feature* is any perceived geometric or functional element or property of an object useful in understanding the function, behaviour or performance of that object.

*Feature-based modellers* simplify design by using features, which are commonly used elements in mechanical engineering such as holes and pockets. When users want to place a boss in a plate, for instance, they specify a height, radius, and centre rather than building every detail of the entire geometry.

*Feature recognition* is the process of identification of expected patterns of geometry, corresponding to a particular engineering functionality in parts of an object

*Filters* are used to reduce the error contained in an input scanned data set.

*Knowledge-based (ruled-based or expert)* systems apply AI technology to capture designers' expertise in computer software so that the system can come up with a design based on specifications fed in by the user. Systems generally contain a

knowledge base of specialised information that the expert uses to do his or her job.

*Learning rate* is used to determine the change of the current weights as ANN is being trained. It is actually the rate of convergence between the current solution and the global minimum in the nature of the error space.

*Manifold* is an object that, at every point on its surface, divides space into exactly two regions, one inside object and one outside the object.

*Maximum likelihood estimator (MLE)* presents the likely errors of the best-fit parameters. The objectiveness of this statistical method is to estimate the parameters of the reconstruction function that most likely correspond to the scanned data set.

*Momentum rate* in an ANN attempts to speed convergence at or near the global minimum by preventing the search from falling into local minima during the search process.

*Object-oriented* programs consist of ready-made blocks and make line-by-line programming unnecessary.

*Parametric Design (PD)* is a method of representing geometric models as a set of shapes, dimensional parameters, constraints, and formulae. Dimensions (such as linear and angular) can be given fixed values or depend on one or more other dimensions through user-defined formulae.

*Process Planning* is to determine the detailed sequence of production steps required to fabricating an assembly from start to finish as it moves from workstation to workstation on the shop floor.



***Product Data Management*** is the technology and methodology used within an enterprise to organise, access, and control data related to its products and manage the lifecycle of those products. A single PDM system can work with CAD, CAM, CAE, other software applications, and with traditional non-computer-based systems that generate or use product data. It also can use a mix of computers, workstations, and associated hardware. PDM system provide access and security controls, maintain relationships among product data items, enforce rules that describe and control data flows and processes, and provide notification and messaging facilities.

***Rapid Prototyping*** systems quickly transform CAD models into small physical models. This gives users the ability to feel, touch, and otherwise handle the parts in the early stage of design. Although vendors use different technologies in their particular equipment, most systems operate using the same basic layering process.

***Reverse engineering (RE)*** enables designers to reuse existing parts and to rapidly capture the mathematical representation of existing forms for reducing the product development cycle.

***Sampling*** is to remove duplicated or redundant points within acceptable tolerance since huge point clouds can not be manipulated efficiently in RE. It needs to be determined precisely how many (or interval) points will be sampled. This amounts to saying that we can not reconstruct original parts correctly if the scanned data is sampled beyond the criterion.

***Segmentation*** is to logically divide the original point cloud set into regions, one for each feature, so that each region contains just those points sampled from a particular feature. Each of these regions is devoid of sharp edges and can be

modelled with a single feature. Extracting-feature can be carried out only in the given segmented region.

*STEP* is an international data-storage standard to succeed and expand upon IGES.

STEP represents data from all aspects of design and manufacturing processes and for different product types.

# Chapter 1

## Introduction

Time-to-market considerations are becoming the key to economic survival in today's global markets. Competition and customer demands are increasing whilst time and resources are often limited. In order to remain competitive, many companies have focussed on improving quality and reducing the time-to-market using concurrent design, rapid prototyping and reverse engineering (RE). The design and prototyping of products in the manufacturing industry often involves the use of physical objects for initial functional and aesthetic design, performance testing and object modification. Hence, a robust methodology is required to acquire exact computer-aided design (CAD) models from an existing part in diminishing lead-times, which meet the functional and aesthetic requirements of the customer at the right cost. Furthermore, quality control (QC) has become one of the most important issues in manufacturing because of increasing demand of higher product quality. Geometric inspection of parts is essential for ensuring product quality.

CAD is regarded as a process that starts with ideas, progresses to geometric models, and then feeds the computer-aided manufacturing (CAM) of real parts. In RE, however, physical parts are scanned first, and then the resulting scanned data are transformed into appropriate CAD models: from which CAM techniques can be used to make more parts.

## 1.1 Definitions of Reverse Engineering

The definition of RE is broad and it encompasses many engineering approaches in which an existing part is analysed either prior to or during the reconstruction process. RE is a process to create CAD models of a physical part from the scanned point data acquired using three-dimensional (3D) position scanners such as coordinate measuring machines (CMMs), laser scanners, optical scanners, and so forth.

The process was defined by many researchers with respect to their specific application. Rekoﬀ, J.R. (1985) described RE as the action that create a set of specification for a part by someone other than the original designer, based upon analysing and dimensioning the part. Abella, R. *et al* (1994) considered RE as the basic concept of producing a part based on an existing model without the use of an engineering drawing. His research was on the direct construction of the geometry using human intervention for feature and relationship definition. Green, P. *et al* (1994) addressed that RE enables designers to reuse existing parts and to rapidly capture the mathematical representation of existing forms for reducing the product development cycle. Sarkar and Menq (1991) also



described RE as a process of retrieving new geometry from the manufactured part by scanning and modifying the existing CAD model. These concise definitions describe specific functions covered within the global term '*reverse engineering*'.

## 1.2 Statement of Problems

Advancements in CAD and CAM related technologies have helped to reduce product introduction costs and lead-times. However, at some stages of design process, designers still need to physically interact with their design for testing the design performances and obtaining an optimal product modification. This means that many design processes strongly rely on physical models, and this is a crucial means of testing a design.

The problem of reconstructing 3D shapes has received a huge amount of attention in manufacturing engineering in the past decade. Most of previous work deals with fitting and machining of free form surface models (Sarkar, 1991; Bradley, 1993; Lai, 1996; Seiler, 1996). On the other hand, very little research work of 3D solid model reconstruction (Motavalli, 1996, 1998; Chivate, 1993, 1999) from scanned point data has been reported. Reconstructing solid models directly from scanned point clouds is of paramount importance since the resulting reconstructed solid models can be directly imported into commercial solid modellers without loss of any geometric and topological information (Thompson, 1995). Once a solid model has been generated, it can be used for a variety of applications such as stress analysis using finite elements,



process planning, integral property computation, assembly analysis, and numerical control (NC) machining. Furthermore, constructing a solid model from scanned point data is useful for the comparison between the original solid model and a manufactured object in solid modelling system (Chivate, 1993).

Prismatic shapes such as *block*, *pocket*, *step*, *slot*, *hole*, and *boss* are very common and crucial in mechanical engineering products. In the geometric modelling view, some researchers (Merat, 1992; Thompson, 1995; Zhang, 1997; Jain, 1998) define above shapes as features. This is because they have a simple representation in terms of a few geometrically meaningful parameters, i.e., length, top radius, bottom radius, centre position, and orientation are enough to describe a cylindrical feature. These applications can be automated in the computer based down-stream activities if a feature-based product description is available (Shah, 1995).

### 1.3 Aims of the Research

The aim of this research is to develop a RE system that can be used to support the entire industrial development of a manufacturing process from conceptual design to manufacturing by capturing more high level semantics. The system will reconstruct solid models of prismatic features directly from a set of scanned points using artificial neural networks (ANNs).

In order to design and develop a generic methodology for the exact RE of components, the following six main tasks were investigated:

- (1) develop a point data reduction algorithm;
- (2) develop edge point detection strategies;
- (3) develop algorithms for deriving geometric attributes from a scanned data set;
- (4) develop a ANN-based feature recogniser;
- (5) develop a feature extraction module from the detected edge points of parts;
- (6) develop a reconstructed CAD model exchanger into other CAD/CAM systems via IGES.

## 1.4 Research Methodology

The overall methodology of the research is depicted in Fig. 1.1.

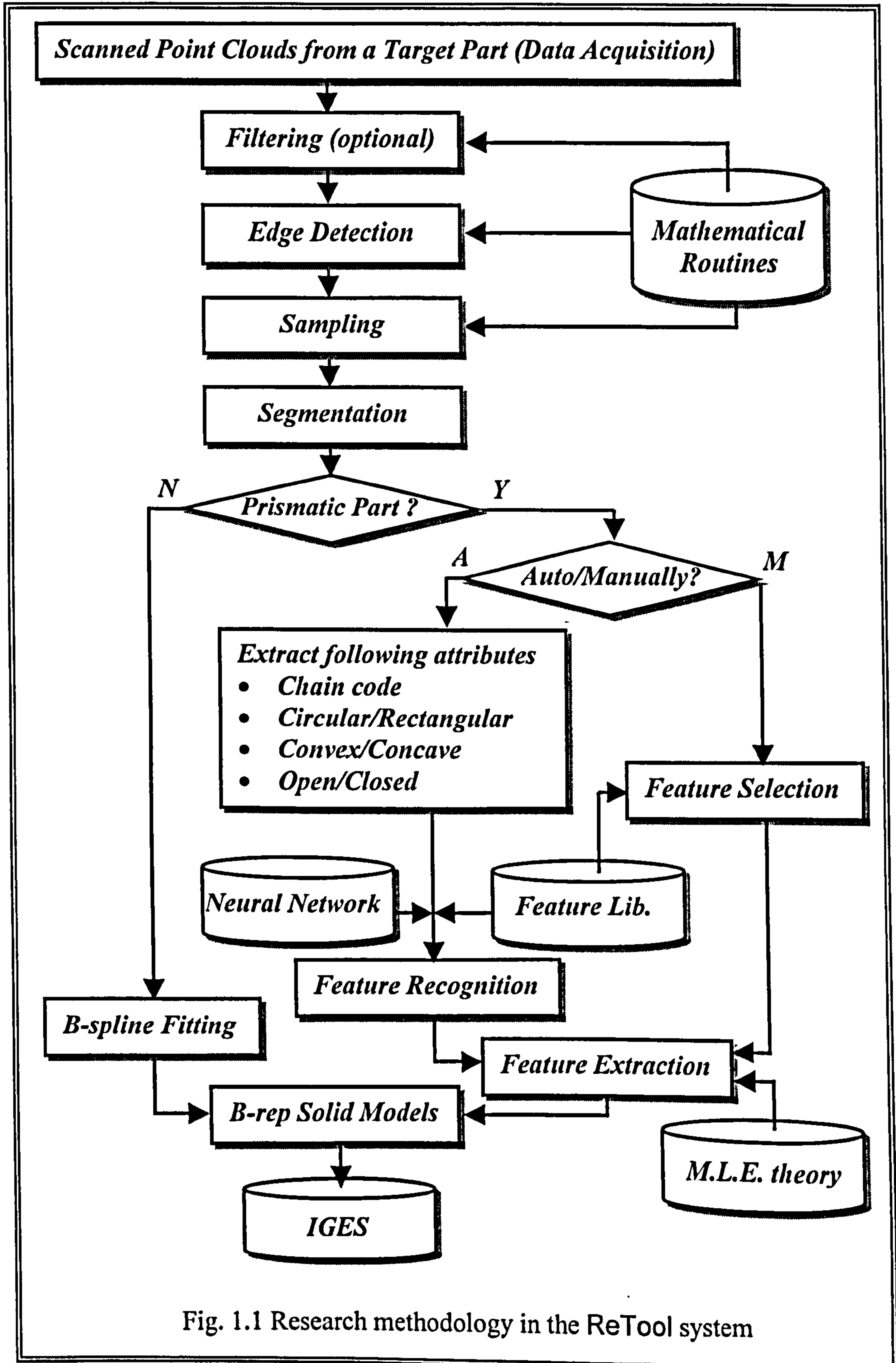


Fig. 1.1 Research methodology in the ReTool system

The overview of individual step is introduced in the following subheadings.

### **1.4.1 Data acquisition and reduction**

A target part is first measured by a scanner using an appropriate scanning strategy, and the acquired scanned points are downloaded to the proposed system, ReTool.

The initial scanned points normally contain many redundant and erroneous points, which are not used for model reconstruction. For the system efficiency, consequently, it is essential to intelligently reduce the data volume by removing measurement error and the redundant points. The point data can be sampled within pre-defined tolerances by the proposed data reduction module.

### **1.4.2 Edge detection**

To reconstruct CAD models directly from a set of point data, a mediator is required as a bridge between the point data and corresponding CAD model. It is because a point by itself is just a discrete 3D entity having no topological relationship to any other points.

In this research work, edge points<sup>1</sup> are detected and adopted as the bridge since it is much straightforward to recognise CAD models from detected edge points rather than from initial raw point data. Therefore, all outer and internal edge points of corresponding features are detected first by developed edge detection

---

<sup>1</sup> Edge points represent either the outer boundary between two different regions in the part with relatively distinct discontinuities, or internal boundaries in a region.



modules. Edge point is a crucial entity in this feature-based RE algorithm since features are recognised and extracted with the detected edge points of the part.

### **1.4.3 Segmentation**

Many products, from household appliances to automobiles, contain separate and distinct regions: both prismatic and free form. Segmentation is to logically divide the original point cloud set into subsequent regions, so that each region can be mapped with a free form surface or a prismatic feature. Segmentation is a vital step towards the interpretation of discrete scanned data since fitting-surface or extracting-feature can be carried out only on the given region. The level of subdivision depends on the complexity of the part. The segmentation of input point data was manually accomplished by RE operators. The mouse is normally used to select enough points on the displayed image.

### **1.4.4 Free form surface creation**

RE has been used extensively in the field of surface model generation. The shape of final products today exhibits an emphasis on free form shape. Whether it is a car, household equipment or injection moulded plastic product, all have free form surfaces in their design. Free form surfaces are being more and more into the design of new parts for many reasons including aerodynamic advantages and improved aesthetic. In this research, a B-spline modelling module has been designed and developed for creating free form surfaces from a point data set.

### 1.4.5 Feature recognition

A CAD system for feature recognition is required to be fast and robust (Buurman, 1990); otherwise recognising features is often error prone and time-consuming. Most research works for extracting geometric features are intended to work with existing CAD models using a graph relationship among vertex, edge, and face. A complete solid model or a feature has never been developed directly from the real scanned data (Thompson, 1995). It is because there is lack of work regarding automatic feature extraction algorithm directly from the imperfect scanned data, so the feature identification is usually left to the RE operator (Chivate, 1993).

A main difficulty in the RE process comes from the lack of geometric information among scanned point data. This is because each individual point does not carry any information beyond pure 3D position coordinates. Points do not contain any additional important geometric information like tangents or curvatures. In addition, points do not capture design history, functionality, aesthetic aspects or the stylist's intentions that would dramatically ease part reconstruction. In this research, a learning process based on the general philosophy of the standard back-propagation (BP) processor are employed to recognise a feature directly from a real scanned data set.

There are two typical benefits to employ ANNs:

- (1) Making inferences from a background of imperfect or noisy point data;
- (2) Learning from examples and generalising this knowledge so that it will apply in new and different circumstances.

ANNs require parallel input patterns. In this research, a group of 2.5D information extracted from 3D scanned points is the given input to the ANN. Each feature in the feature library is uniquely described by four geometric attributes: *chain codes*, *convex/concave*, *circular/rectangular* and *open/closed parameter*. Identifying each feature requires the determination of these attributes. The primary features in the feature library are classified based on these attributes. If the attributes are determined, a given subpart may be uniquely identified as a valid library feature. New and robust algorithms are developed for determining these attributes for each of the features.

#### **1.4.6 Feature extraction**

If the user selects some portion of the scanned data that can be reconstructed into a surface or a feature, the system automatically create the corresponding CAD model with pre-trained ANN module and MLE theory. During reconstruction, the user can modify the dimension of extracted features by varying the feature parameters under the *parametric design* scheme.

There have been some attempts to extract quadric entities from random point data. Among them, the most reliable method is based on the MLE theory (Bolle, 1986). In this research, the MLE theory is used to extract the dimensional data of recognised features.

#### **1.4.7 CAD model exchange**

There is increasing interest in performing design, manufacture, analysis and planning concurrently in order to cut down the product development cost. Thus,



there is a need to share product data between the various commercial applications used during design and manufacturing processes.

The reconstructed CAD models stored in the database of ReTool system are compatible with the initial graphics exchange standard (IGES) data file format, so that output model file can be exchangeable with other applications. Using this approach to the data format gives more flexibility to various kinds of commercial systems.

## 1.5 Assumptions of the Research

Since RE is a rather complex process, a set of assumptions has to be established.

The fundamental assumptions of the research include:

- The input scanned point clouds are structured data, not irregularly spaced data.
- The accuracy of scanner must be within acceptable tolerances.
- The part is composed of some prismatic engineering features that can be defined.
- No excessively entangled compound feature exists in the part.
- Non-manifold<sup>2</sup> objects are ignored in this research.

---

<sup>2</sup> A *B-rep* based model satisfying the *Euler's law* with topological relationship is *manifold*, otherwise *non-manifold*. That is, real world physical objects are *manifold*. More detailed explanation is presented in Chapter 5.



## **1.6 Thesis Outline**

In the following chapter, a brief survey of previous related work is presented in order to introduce the state-of-the-art RE process and to put the problems involved into perspective. In chapter 3, filtering and sampling modules developed are discussed and the novel method for edge detection is described. Chapter 4 outlines B-spline modelling techniques to reconstruct free form parts. The basics of the B-rep and feature-based modelling techniques implemented in the application are presented in the chapter 5 and chapter 6, respectively. The detailed feature recognition algorithms using ANNs are explained in the chapter 7, and a pre-processing method for deriving attributes is also precisely discussed. Chapter 8 describes edge extraction methods using MLE theory, and developed IGES neutral file exchanger is introduced and discussed. The chapter 9 presents a prototype implementation along with some examples and their experimental data. Finally, the chapter 10 presents the general discussion of this research with conclusions and suggests further research activities.

## **Chapter 2**

# **Review of Reverse Engineering Technology**

In the process of RE, an original part needs to be systematically analysed to the extent that a replica can be created precisely. There are two distinct specifications that are required to undertake RE efforts for the purpose of reconstructing a replica: functional and dimensional (Rekoff, 1985). The functional specifications describe how the original part works. A crucial use of the functional specifications is to guide the adjustment of a product during manufacture. In contrast, dimensional specifications establish the part dimensions – lengths, angles and tolerances.

In this research, dimensional specifications were focused on since the shape and size of the part is the most important factor in RE processes. The purpose of this chapter is to describe the state-of-the-art RE processes and to identify the problems that require further research.

## **2.1 Why use Reverse Engineering?**

The existence of a CAD model provides enormous profits in improving the quality and efficiency of design, manufacture and analysis. The RE technology is crucial in the following circumstances (Yau, 1997):

- CAD systems were not applied in the original design;
- CAD models are unavailable for parts that must be reproduced;
- Existing parts are modified due to manufacturing limits or design change;
- Documentation on the original design would not be enough to support further modification;
- A worn or broken part that is no longer available from the original supplier often needs to be replaced with new one (Rekoff, 1985; Thompson, 1995).

In general, a product development process from initial conceptual design to final product is an iterative process that includes product design, analysis of performance, product prototyping for experimental evaluation and design modification. It is always started by designer's concept based on the specified functional and aesthetic requirements, in which the aesthetics of a product is sometimes taken a prerequisite than functionality.

The initial aesthetic design of parts is often carried out by designers who make a physical model using wood, foam, clay, and so forth. After creating a prototype, engineers can assess their design validity with it. From that point of view, prototyping is still widely used to test the design of parts in industry.



A review of the feedback from industry today shows that RE is a vital technology in the following manufacturing business. The common use of RE is available in the automotive industry where the style of the interior/exterior design can be ascertained through a touch-probe scanner with the clay models. Stylists still usually use full-scale clay models due to the problems of working at small scale with 2D computer screen. The development of a new car takes a shorter time from concept to prototype models with the RE technology. If a clay model is created, the model must be scanned again, and its CAD model can be reconstructed with the new scanned data set. This iteration occurs several times during the design stage.

In the aerospace industry, thousands of sheet metal parts are used to assemble an aeroplane, and they often need to be reproduced or modified. In the case where an old aeroplane has to be redesigned, the CAD models of the existing parts are often created by scanners (Huang, 1994).

In die and mould manufacturing, the existing design is often modified on the shop floor due to manufacturing limitations, and up to 80 ~ 90% of surface machining is still carried out by copy milling (Werner, 1993).

Another application area, which may be becoming more crucial in the future, is the generation of custom models for use in manufacturing just a single product. For instance, each human being is not identical: scanning individuals and modelling them for manufactured products such as custom gloves, shoes, and helmets in order to fit exactly. Figure 2.1 shows the England and Arsenal goalkeeper – *David Seaman* – having his hands scanned to generate data for a



pair of goal-keeping gloves. The process begin with a Wicks and Wilson optical scanning device (Triform, 1998). It captures all the data necessary to produce a 3D computer model of the hand. This image is then processed using a RE technique that analyses the millions of data points<sup>1</sup>.



Fig. 2.1 Scanning hands to making goal-keeping gloves  
(Image courtesy of Warwick Manufacturing Group)

Hence, if RE technology is appropriately applied in manufacturing industry, it can be a powerful tool for enhancing the communication between customers and suppliers in a concurrent engineering environment.

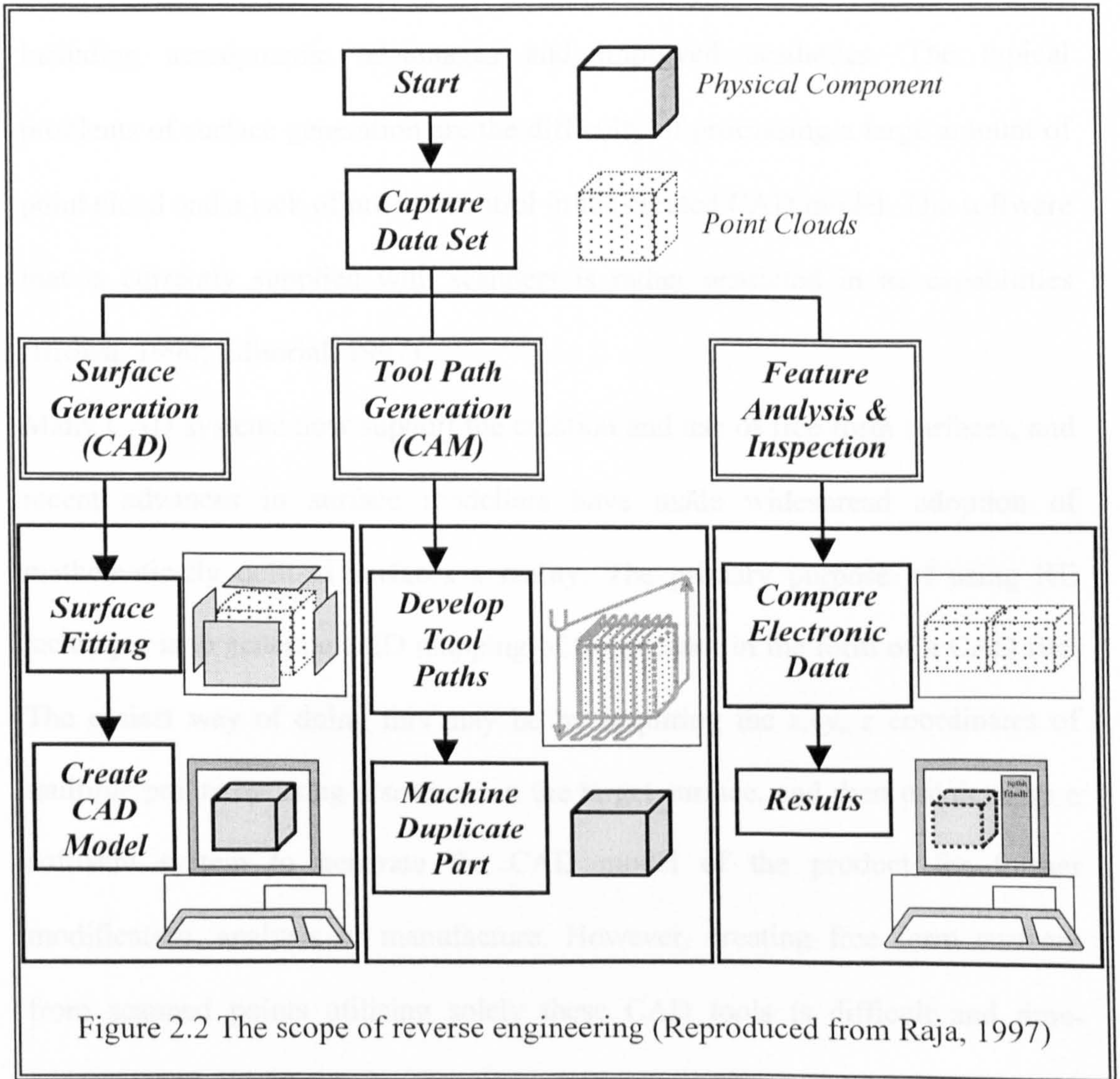
---

<sup>1</sup> The researchers on the glove project, led by Vinesh Raja (principal research fellow at *Warwick Manufacturing Group*), have been working with *Umbro's* own Research and Design team.



## 2.2 The Scope of Reverse Engineering

The RE functions can further be divided into three branches as shown in Fig. 2.2: surface generation, tool-path generation, and feature analysis and inspection.





## 2.2.1 Surface generation

### 2.2.1.1 Surface generation in RE

The process methodologies for RE have received extensive attention, especially on the field of surface model generation. Free form surfaces are being increasingly incorporated into the design of new parts for many reasons including aerodynamic advantages and improved aesthetics. The typical problems of surface generation are the difficulty of processing a large amount of point cloud and a lack of precise control in the created CAD model. The software that is currently supplied with scanners is rather restricted in its capabilities (Brown, 1990; Editorial, 1997).

Many CAD systems now support the creation and use of free form surfaces, and recent advances in surface modellers have made widespread adoption of mathematically defined surfaces a reality. The primary purpose of using RE technique is to generate a 2D mapping of the product in the form of a CAD file. The easiest way of doing this may be by acquiring the  $x$ ,  $y$ ,  $z$  coordinates of multiple points by using a scanner on the target surface, and then use/develop a software system to generate the CAD model of the product for further modification, analysis or manufacture. However, creating free form surfaces from scanned points utilising solely these CAD tools is difficult and time-consuming due to the lack of accurate interpolation algorithms to describe the point cloud into an efficient 3D information. It requires considerable skill, and usually takes several iterations with testing of physical prototypes (Green, 1994). That is why surface generation is still one of the most popular research issues.

### 2.2.1.2 Previous work

Interest in fitting methodologies for surface interpolations is still very high for researchers. Cubic and bicubic surface algorithms were proposed for RE of composite sculptured surfaces by Seiler (1996) and Lai *et al* (1996). They employed the constrained least-square fitting techniques for converting uni-directional scan data into spline curves. The splines were then blended to a surface model by a spline segment matching and subdividing algorithm. A model refinement method was also proposed in their work. Milroy, M. *et al* (1996) proposed a multiple patch RE algorithm to model parts where many types of surface patches, such as quadric and free form, were blended together on a part. Bradley, C. *et al* (1993) examined several methods of modelling free form surfaces from irregular data. In their paper, the focus was on the methodology and algorithms developed for reconstructing curved surface patches. Semi-automated surface generation method was proposed in (Lazar, 1995). For obtaining a smooth parametric surface approximation with multiple surfaces, Sarkar and Menq (1991) used the least-square fitting of B-splines. Menq and Chen (1996) proposed a method that used a series of cross-sectional curves to construct the surface model of the part through a set of ordered points generated by a CMM. Since this method was to use the iterative measurement procedure to diminish error caused by compensation for the probe radius, it was too slow. Moreover this method can be applied only for smooth surface fitting.

### 2.2.1.3 Surface modelling algorithms

In the reconstruction of the surface, local fitting and global fitting can be used. For *local fitting*, the part surface is divided into regions through segmentation,



and then surface-fitting is applied only on the given segmented region. Surface-fitting is simple but segmentation is difficult. For *global fitting*, the part surface is interpolated into one surface. This avoids the problems of surface position and tangent continuity but huge amounts of point data need to be processed.

There exist various surface patch models, such as parametric polynomial surface models, boundary interpolating surface models, or sweep surfaces which can be used to create complex surface shapes. In applications of RE, parametric polynomial surface models and boundary interpolating surface models are most common. Typical examples of parametric polynomial surface models encompass the Ferguson surface, Bezier surface, B-spline surface and non-uniform rational B-spline surface (NURBS). The algorithm of B-spline based fitting is given in chapter 4. Boundary interpolating surface models, including a ruled surface, lofted surface and Coons surface, are also widely adopted in RE. These surfaces are constructed by connecting several cross section curves.

### 2.2.2 Tool-path generation

The second branch of RE function is to generate tool paths. Scanning of an existing part provides enough data to create tool paths and machine an exact copy, without generating a CAD model, if desired.

An example of the application of tool path generation from a point cloud set is provided by Renishaw's *Tracecut<sup>TM</sup>* software (Renishaw, 1998). In Fig. 2.3, the output NC tool paths are generated directly from the scanned point data. This is



the simplest and most widely employed method of handling the scanned data to produce an exact copy of the part.

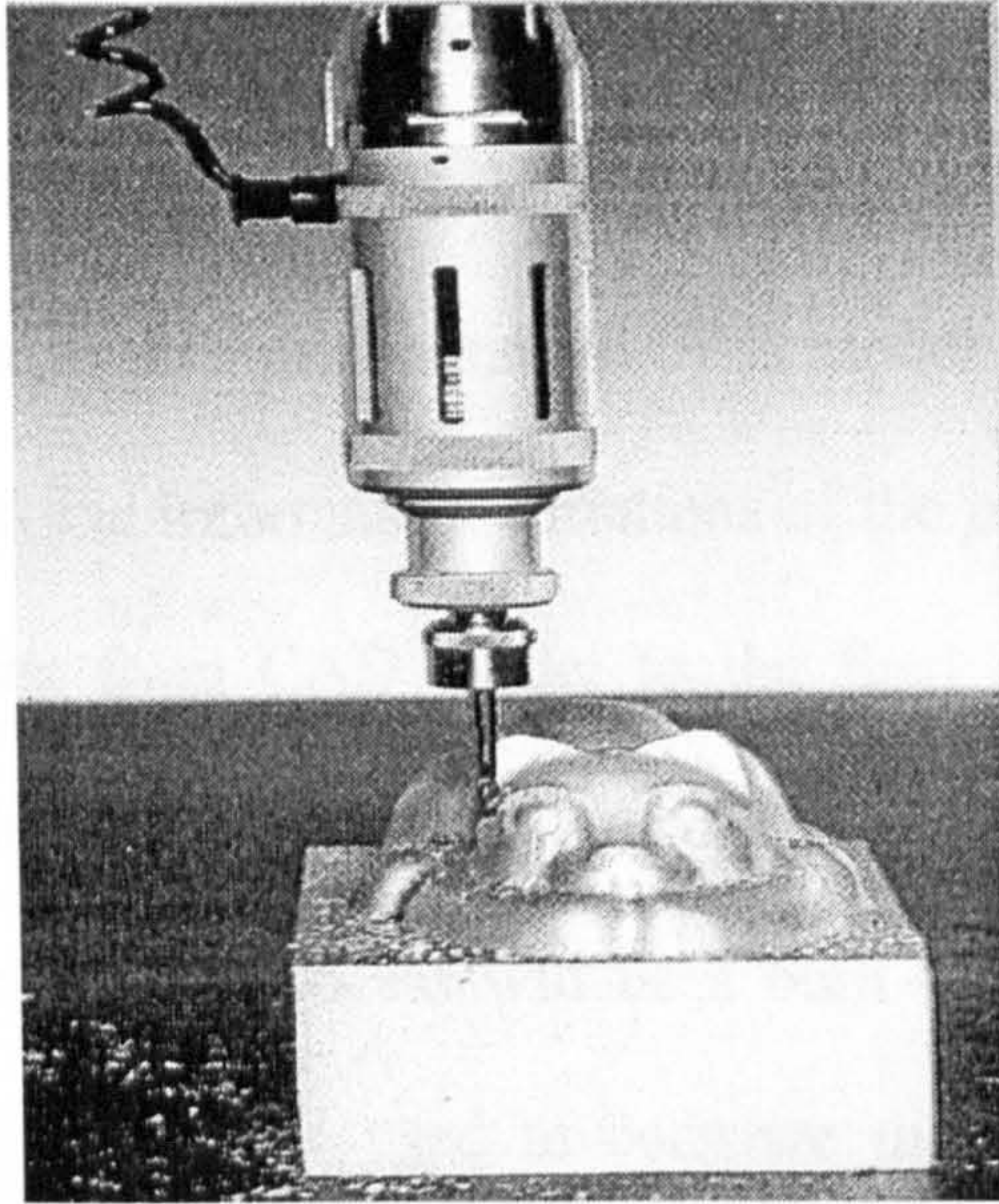


Figure 2.3 Tool path generation from scanned data  
(Image courtesy of Renishaw)

### 2.2.3 Feature analysis and inspection

The last branch of RE function is to deal with features for improving design performance. Whether the design process is driven by the aesthetics or functionality of a part, engineers can interact with prototype models and make changes based upon their implicit knowledge and/or empirical data. Once the engineers are satisfied with the changes made, the information describing the new features of the part must be captured - in electronic form - in order to update the working model (Raja, 1997). RE for this option is referring to the



comparison of the point cloud files which describing the physical entities of the part.

During the design phase, nominal CAD models are constructed using CAD applications. These nominal CAD models represent all the geometric details of the part and its components. These models are the desired end product, however, in practice, because of the variations in manufacturing processes, the end results is not exactly the same as the initial intended CAD model.

To solve this problem, the information variations of the part need to go through a number of iterations from CAD model to the final product. Within these iterations, a certain level of accuracy tolerances is accumulated. Furthermore, the end result of each CAM process will be a built up of tolerance from the previous operations. RE can be used to compare the final product with the original CAD model. As a result, continuous monitoring on CAM process can be done to ensure that right components are produced. That is, products that shrink, twist or in some way deform from their initial intended shape can be scanned and adaptation made to the design or to the manufacturing process that result in the desired shape.

This option can help designer to test the design of parts by comparing initial data with the reconstructed CAD models of the final product. The results of the comparison can be used to develop the corresponding manufacturing process.

## 2.3 Data Acquisition

The first step of the RE process is to capture geometric information representing the physical component shapes. Over the years, many 3D scanners have been developed to scan the surfaces of the target part, producing a point cloud, which as a whole define the surface shapes. Due to the almost infinite range of shapes to be scanned, with great differences in requirements concerning their accuracy, a great variety of scanning methods has been developed.

These scanners fall into two distinct categories: *contact* and *non-contact* device.

*Contact* method uses mechanical probes for interacting with the surface of the part. The most common uses of contact method in industry are CMMs and touch trigger machines. They consist of an arm or a gantry mechanism with a touch probe mounted on the arm. They collect data by using an analogue constant contact probe which produce a series of data points based upon the deflection of the stylus tip (Galt, 1997). During the scanning process, the system records the physical entities of a component in the form of numerical data. This data is in the form of a point cloud file, with each point representing a  $x$ ,  $y$  and  $z$  position.

A more recent addition to this technology is the *non-contact* scanners which have evolved with the machine vision industry, such as laser, optics and charge-coupled device (CCD) camera. These probes typically use triangulation to measure range and acquire a single point at a time, similar to the standard probe.

A further advance is the use of a scanning technology that provides a line on the part surface with a large amount of point clouds resulting in a faster and more detailed digitising methodology. Other benefits are also derived from this



approach as the density of data collected allows image display and analysis techniques to be used to show such as the shaded image of a part. These scanners can digitise unknown surface and they can produce large amounts of point data in a very short time period without touching with the target part. In recent years non-contact scanning technology has been widely investigated (Bradley, 1993; Hosni, 1994; Huang, 1994; Milroy, 1996; Chen, 1997b; Motavalli, 1998; Chua, 1998).

Some RE projects used contact and non-contact device simultaneously. Chen L. *et al* (1997b) employed a CMM machine equipped with a touch-triggered probe and a computer vision system. A CCD camera captures the global information from the component surface, whilst the contact probe extracts accurate coordinate data. This kind of cooperative sensor integration can be used to enhance CMMs by obtaining much higher digitisation effectiveness. While all these systems produced 3D point data, they vary widely in the point densities and accuracy ranges they provide. The format of scanned point data by a contact device is always one-way directional, but non-contact devices often generate randomly spaced point data. Each method has strength and weakness of its own and hence the selection of 3D scanner must be considered carefully, dependent on the speed and accuracy.

To sum up, all scanning methods must interact with the surface or internal material using some phenomenon such as light, sound, magnetism or physical contact. The following section describes scanning devices in depth. The detailed list of the commercially available 3D scanners with their accuracy, speed, and price is given in Appendix 1.

## **2.4 Scanning Devices**

### **2.4.1 Contact scanning devices**

#### **2.4.1.1 Manual measurement**

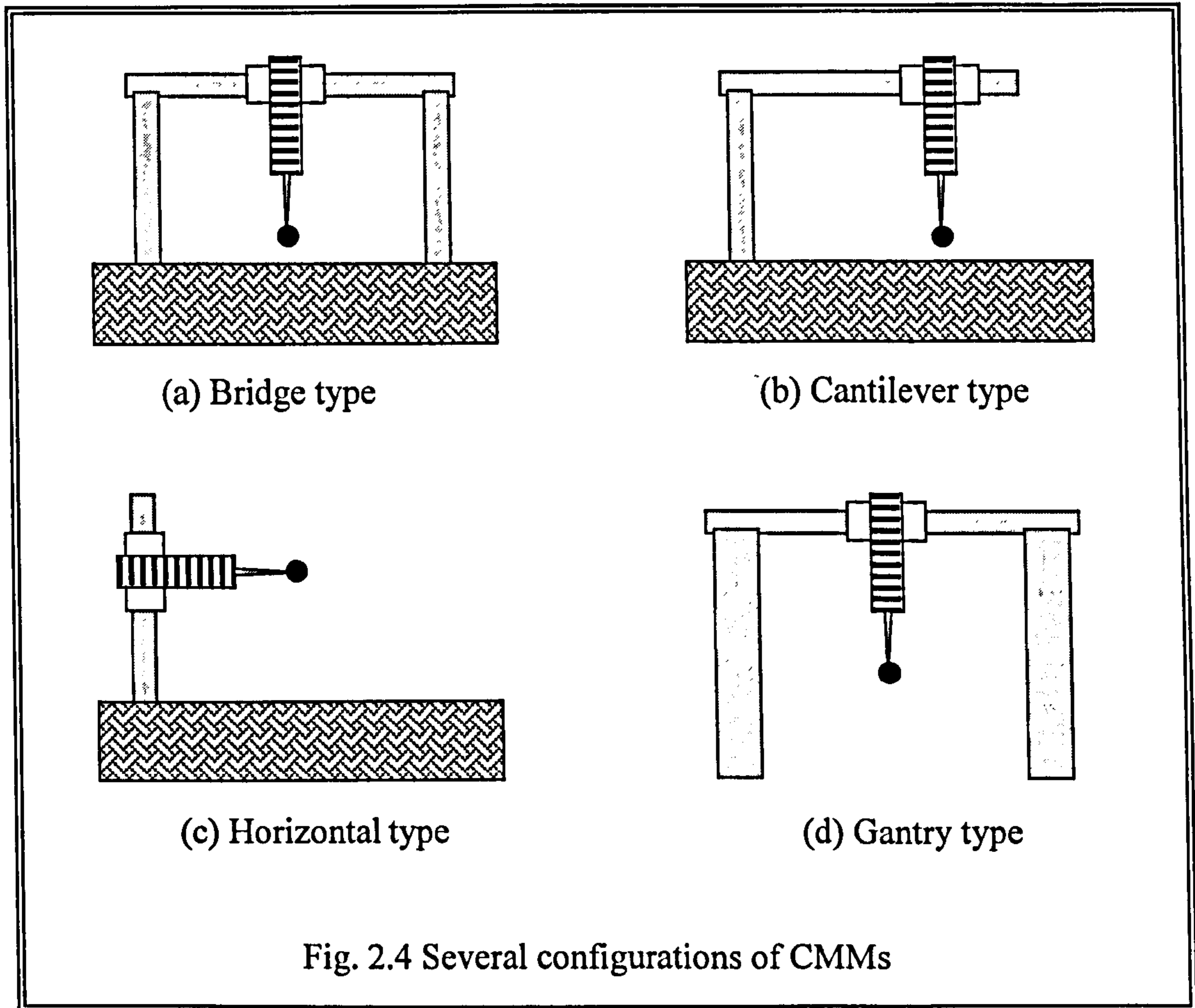
In this technique, some typical points across the surface of a part are identified and the  $x$ ,  $y$ , and  $z$  coordinates are manually measured using vernier, callipers, measuring gauge and blocks. From the available coordinates, a 3D CAD model that consists of curves and surfaces is generated to interpolate these points. This method requires intensive effort, both for the data measurement and data entry. The accuracy of this method depends on the number of measured point and the precision of the measuring tools. Although this technique is still used in some small companies, it is being pushed out.

#### **2.4.1.2 Coordinate measuring machines (CMMs)**

A CMM is a 3D scanning device based on three orthogonal axes and employ a contact probe to scan the surface of the part (Brown, 1990). These axes, designated  $x$ ,  $y$  and  $z$ , embody a spatial coordinate system, similar to that of a NC milling machine. A CMM can have a number of different configurations, including cantilever, bridge, horizontal arm, and gantry as shown in Fig. 2.4.

It consists of a mount which can hold different probes which can move in any of three coordinate directions. CMMs have three movable components along three mutual perpendicular guideways. In the bridge type configuration, the probe is attached to the first component, which moves vertically to the second. The second component moves horizontally relative to the third. The third

component is supported on two legs which move horizontally relative to the machine base. No rotations of the probe are possible.



The probe is a sensitive pressure-sensing device that is triggered by any contact with a surface. It automatically follows the contours of a physical surface, then the linear distances moved along the three axis are recorded by length-measuring systems, thus providing the  $x$ ,  $y$ , and  $z$  coordinates of the point (ASCII data). The sets of  $x$ ,  $y$ , and  $z$  points are then processed to create substitute geometry entities that represent the part's real geometry. A fundamental problem with this device is that it requires compensation of the touch-probe radius.



The scanning accuracy of CMMs has improved significantly since their first advent in the early 1960s. Although the process usually takes a lot of time when applied to scanning free form parts, CMMs are often crucial tools in many precise inspection systems. Positional accuracy on the order of  $\pm 3$  microns locally and  $\pm 14$  microns corner to corner is possible (Thompson, 1995). Fig. 2.5 shows a contact scanning device, based upon CMM technologies.

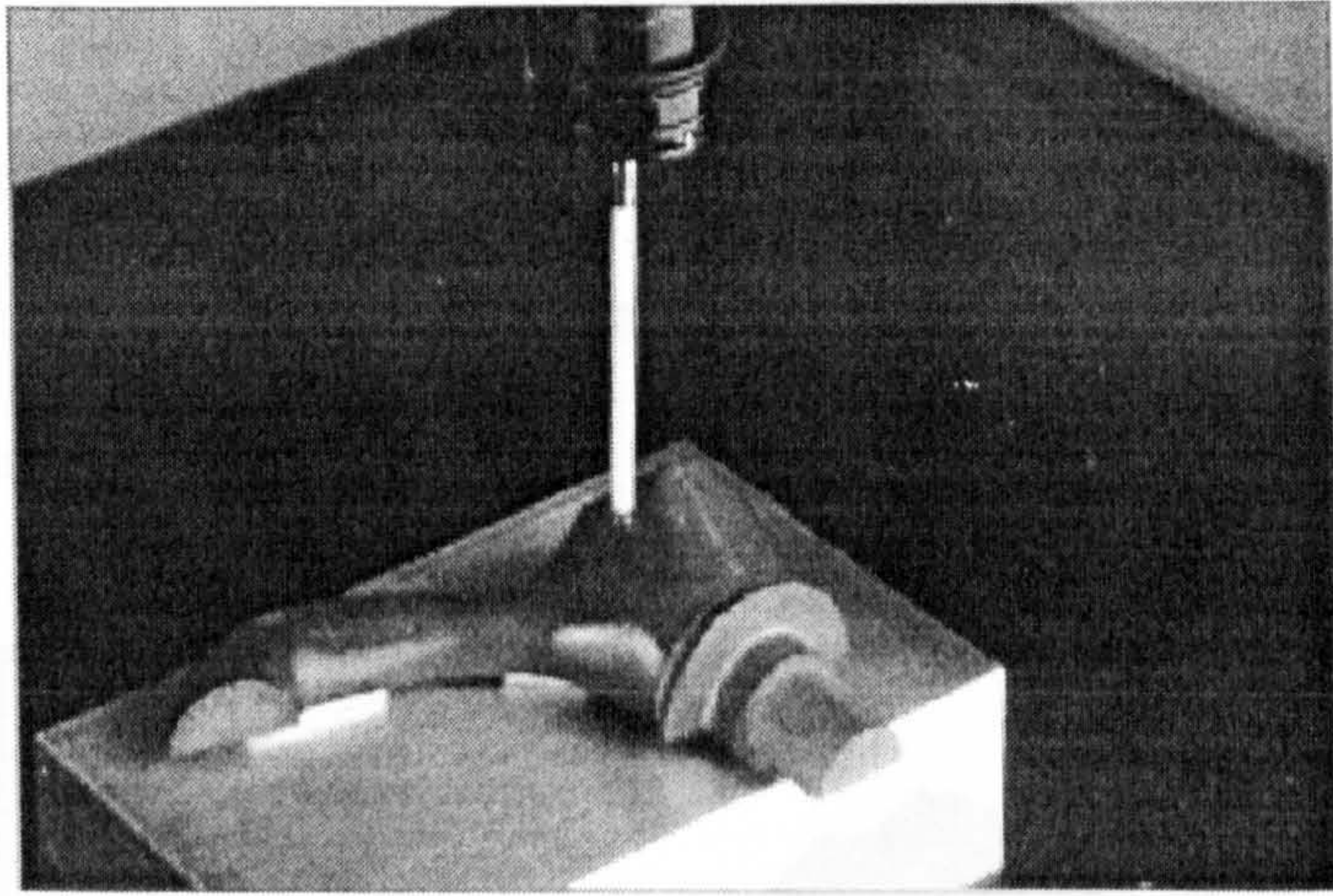


Figure 2.5 Contact scanning device  
(Image courtesy of Renishaw)

Figure 2.6 Spot ranging scanning device

The  $x$ ,  $y$  coordinates and the range of the surface point is given by the location of the source and analysis of the reflected beam, respectively. The  $z$  coordinate of the point can be derived from the phase difference between the incident and



## 2.4.2 Non-contact scanning devices

Non-contact scanning devices usually use laser range finders, moire interferometers, stereo image finders or structured lighting devices to capture point data. Since scanned point clouds are captured without physically contacting with the part, non-contact scanners are less accurate than CMMs. The positional accuracy normally exceeds  $\pm 50$  microns [Raja, 1997].

### 2.4.2.1 Spot ranging

The principle of the spot ranging is to discharge a beam onto the surface of the part, and then follow the reflected beam using a sensor that is placed coaxial to the beam source as shown in Fig. 2.6.

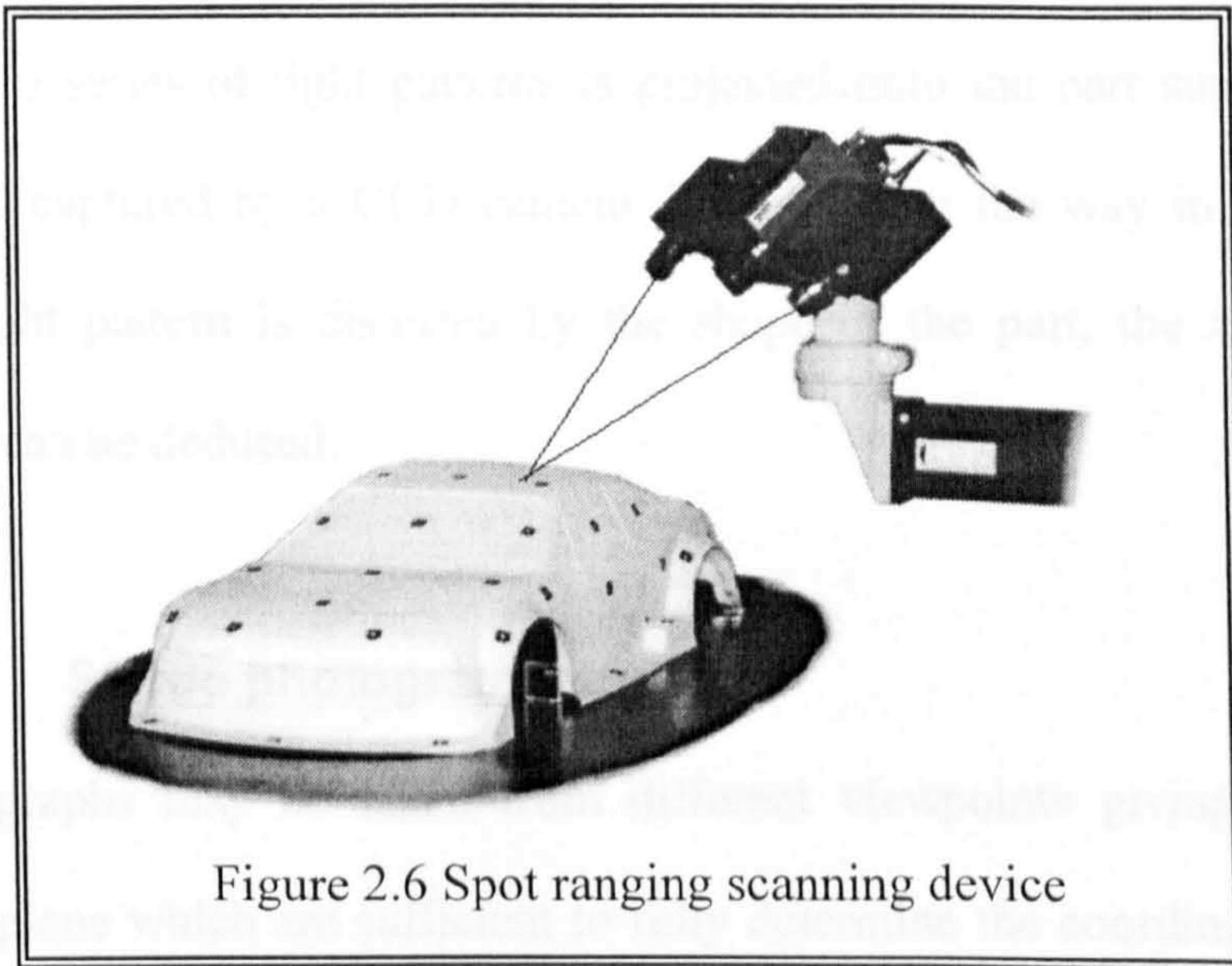


Figure 2.6 Spot ranging scanning device

The  $x$ ,  $y$  coordinates and the range of the surface point is given by the location of the source and analysis of the reflected beam, respectively. The  $z$  coordinate of the point can be derived from the phase difference between the incident and



reflected light or by the time taken for the light to reflect back from the part surface.

### **2.4.2.2 Moiré interferometers**

Moiré topography is a technique in which patterns of light, after passing through optical lenses, are used to illuminate the part surface. An interference pattern is projected onto a surface creating lighted contour lines. These lines are captured in an image and then are analysed to compute distances between lines. These distances are proportional to the height of the surface at the spot of interest and so the coordinates of surface points can be computed.

TriForm 3D Image Capture System (TriForm, 1998), from Wicks and Wilson Limited, is a typical example of a scanner using the moiré fringe technique. In this system, a series of light patterns is projected onto the part surface to be scanned and captured by a CCD camera. By analysing the way in which the projected light pattern is distorted by the shape of the part, the  $x$ ,  $y$ , and  $z$  coordinates can be deduced.

### **2.4.2.3 Stereo photogrammetry**

Two photographs may be taken from different viewpoints giving projected points in a plane which are sufficient to fully determine the coordinates of real spatial points. The readings for the tilted positions are later transformed by a standard computer routine to analytically restore the observed surface to its natural position in *Cartesian* space. The procedure of observing and recording

two projections and their subsequent quantitative analysis is called stereo photogrammetry (Duncan, 1983).

#### **2.4.2.4 Structured lighting**

The general methodology for structured lighting scanning is using a sensor, such as a camera, to detect the position of a reflected laser beam from the surface of the part. The image of the part reveals the 2D coordinates of the part surface point. Triangulation procedures are then used to compute the depth of the point, which is located at the intersection of the line source and the line of camera axis (Bidanda, 1994). By sweeping the light source over the part surface, the whole shape of the part can then be scanned.

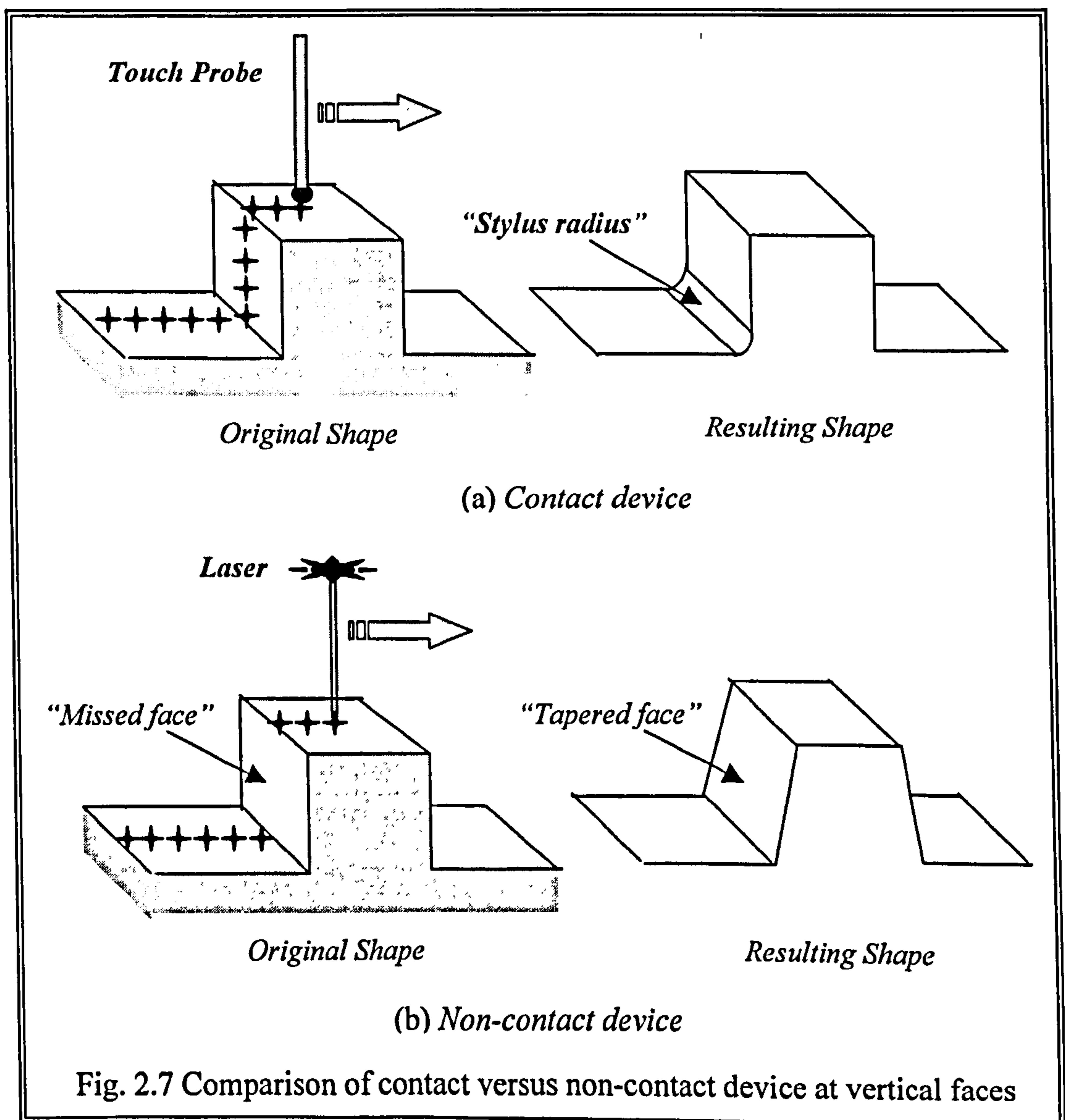


### **2.4.3 Comparison of contact and non-contact scanning method**

Contact probe scanning devices based upon CMM technologies provide an accurate and flexible method for data collection. Since they are robust to operate under shop floor conditions, they are commonly used to collect the point data for subsequent steps of the RE process. In the current market place contact probe scanning devices offer an accuracy which surpasses that achieved by non-contact techniques. The accuracy can be within microns, or even in the sub-micron range. They have now become an integral part of the inspection system in many industries. The slow rate of data capture is, however, a significant shortfall. Also the material of the measured part may limit the use of contact devices because the probes must deflect in order to register a point. In automatic measurement, a precise scanning path must be well designed in advance such that any collision can not occur during scanning.

Compared to CMM touch probes, non-contact devices have the advantages of speed and immediate record of the scanned data into a digital computer file. These devices capture huge amounts of data in a relatively short period of time. However, the primary limitations of these kinds of devices are cost and accuracy. The accuracy is somewhat limited; tolerances are as poor as 20 times achieved by CMMs. It is difficult to obtain a commercial usable scanner within 10 microns (Zhang, 1995). The captured points are often irregular and unformatted. The huge amount of scanned data acquired is not easily handled by most current available CAD/CAM packages. To obtain smooth and accurate CAD models, extensive processing is required to reduce the number of points in

the clouds. Otherwise, the CAD models reconstructed from these scanned data may lose their geometric characteristics and become erroneous for next product development (Chen 1997b). With some vision systems, scanned parts with mirrored surfaces must be sprayed with a paint or powder to reduce light reflection. Some non-contact scanners have problems generating data describing surfaces which are parallel to the axis of the laser (Raja, 1997). As shown in Fig. 2.7 (b), for the laser scanners, there is the possibility that the laser light beam can not reach a part surface.



It is because the inclination angle between a laser beam and a part surface is large, so the resulting surface is regarded as a tapered surface due to a shadow effect. Furthermore, as laser sensing is sensitive to shiny and dark surfaces, a complicated lighting technique may also be required (Menq, 1995).

Deep opening and concavities can also present a problem. If the opening to the surface is smaller than the angle created by the triangulation process of laser scanners, the laser and sensors can neither project nor see the light. The result is missing data in the surface. Detailed comparisons between contact and non-contact method are listed in Table 2.1.

Table 2.1 Comparison of contact and non-contact scanning method

<i>ITEMS</i>		<i>CONTACT</i>	<i>NON-CONTACT</i>
<i>General</i>	<i>Speed</i>	<i>Slow</i>	<i>Fast</i>
	<i>Accuracy</i>	<i>High</i>	<i>Low</i>
	<i>Cost</i>	<i>Low</i>	<i>High</i>
	<i>Calibration</i>	<i>Unnecessary</i>	<i>Necessary</i>
<i>The nature of part surface</i>	<i>Surface treatment for avoiding light reflection</i>	<i>Unnecessary</i>	<i>Necessary</i>
	<i>Part surface hardness</i>	<i>Dependent</i>	<i>Independent</i>
	<i>Scan a vertical plane?</i>	<i>Yes</i>	<i>No</i>
<i>Output points</i>	<i>File size</i>	<i>Small</i>	<i>Big</i>
	<i>File format</i>	<i>Regular</i>	<i>Scattered or Regular</i>



## **2.5 Current Issues associated with Reverse Engineering**

### **2.5.1 Processing a large amount of scanned data**

In data acquisition, the number of scanned points collected is often more than necessary, especially with non-contact method. A good data reduction algorithm is therefore essential to achieving reasonable processing time for CAD model reconstruction in the next step.

The curvature distribution on the surface is a crucial factor for determining the amount of required scanned data. If the scanned point avoids redundancies in uniform and smooth area such as prismatic parts, it is a desirable distribution. Chen *et al* (1997b) has addressed a digitising method that part digitisation process continues until the digitising accuracy satisfies the required tolerance.

### **2.5.2 Automatic extraction of 3D CAD models from 2D data**

Generating 3D CAD models from 2D information has been studied (Ballard, 1996; Shum, 1997; Sheng, 1998; Motavalli, 1998) for many years. Several algorithms have been proposed for the automatic generation of 3D CAD models from given 2D engineering drawings. Those algorithms used three or four conventional orthographic views to generate 3D solid objects that made up of planar surfaces. The reconstruction procedure used a volume-based approach and views a complex part as composed of a number of primitive instances. Ballard, D. (1996) designed a feature recognition system to extract features from the orthographic image captures from the component. These views were then used to compare instances of features with a corresponding set in the feature

library in order to match them. The recognised form features and their relationship were used to develop a feature-based model for the part. To extract the boundaries of the object in each view, this system used Hough Transform (HT) algorithm. The HT converted the difficult global boundary detection problem in image space into a more easily solved local peak detection problem in parameter space. All the views were then compared with a corresponding set in the feature library in order to match them.

Shum, S. *et al* (1997) proposed a 3D solid reconstruction method from 2D line drawings of six orthographic opaque views, viz. top, front, left, right, bottom and rear views. Firstly, the six views were grouped into three pairs. For each pair of views, segmented areas from one of the two views were incrementally extruded according to the information in the neighbouring view. Extrusion primitive solids were generated during the incremental extrusion. All primitive solids were then unioned into an extrusion solid. Finally, all extrusion solids were intersected to give a unique 3D solution object. Motavalli, S. (1998) proposed an approach for automatic 3D modelling of an object using intensity image. With stereo matching technology, 3D wireframe models of the component were created with various orthographic views.

### **2.5.3 Automatic extraction of 3D CAD models from 3D data**

There is lack of work regarding automatic feature extraction algorithm directly from scanned data. Many research works so far for extracting feature have been intended to work with existing CAD models.



Thompson, W. (1995) addressed a feature-based RE technique, but the system completely depends on the user intervention. Motavalli and Shamsaasef (1996) proposed an object-oriented database structure for RE system and feature-based design using orthographic view.

In addition, 3D images and prototypes have recently drawn attention from researchers for diagnosis and treatment in the medical field. A prototype serves as a hard copy of the existing part and provides both visual and tactile documentation for surgical planning and prosthesis design (Chua, 1998). A RE approach can be readily applied in such applications where the complex and highly unique human surface must be recreated in order to obtain an exact fit for a replacement joint, or where a reduction of surgery time can be exploited if the surgeon can physically examine a replica of the anatomy that is to be repaired beforehand (Green, 1994).

Although a fully automatic RE system that can make decisions and classifications is ideally necessary, at the current state-of-the-art, user interventions are still crucial way for part identifications. For a human carrying out RE, it is reliable to recognise whether a part has pocket or not. For instance, suppose an engineer is given a set of noisy scanned points from pocket surfaces. If the engineer is told that a pocket surface was scanned, he or she may successfully reconstruct a pocket feature using pocket parameters. However, if the engineer does not know, this pocket is unlikely to be recognised, and the reconstructed CAD model may be represented as some free form surface element (Varady, 1997). Consequently, feature identification is currently left to



the RE operator. However, it is still not straightforward for him or her to recognise features (Chivate, 1993).

#### **2.5.4 Edge detection**

Medoi, J. *et al* (1987) discussed the problem of identification of 'corner' in digital curve and discusses various strategies for finding corner points. The position at which a point set needs to be divided into two separate subsets of data points can be found from the variation of the curvature of the spline curve, which is defined by the data points as control vertices. Chen, L. *et al* (1997b) suggested a method to detect boundary edges of free form surfaces using *Laplacian* operations. The surface edges and surface creases were detected by *zero-crossing* edge detection method. However, in order to reduce the negative effect of excessive noise from this operator, it is necessary to employ adequate noise-filtering methods prior to edge detection.

#### **2.5.5 Noise and missing data**

Noise usually exists in scanned data sets and it is typically difficult to be removed. It occurs in a variety of ways such as extraneous vibrations, specular reflection, and environmental conditions. Even though a filtering module is used to eliminate the noise, some noise can not be removed clearly. In addition, a filtering process will also reduce the 'sharpness' of the data, then sharp edges are usually replaced by dull edges. This is sometimes acceptable; however, it may cause serious problems in detecting edges in subsequent RE process.

Another serious problem is restoration of missing data during data acquisition. Data that close to the sharp edges are fairly easy to be missed due to the nature of non-contact as well as contact scanning. In practice, the part will have to rest on some surface, so part of it will not be accessible to the scanner. In addition, if the scanner is fixed in position, at any one time, it will be able to capture data from an even more limited region of the part's surface. Thus, in general, it will be necessary to combine multiple views taken with the part placed in different orientation in front of the scanner. The simplest approach is to place the object on a turn table.

### **2.5.6 Data exchange**

One of the most crucial and primary factors in system evaluation would be related to data import and export. Reconstructed CAD models need to be transferred into other commercial CAD systems by standard neutral file formats such as IGES or the STandard for the Exchange of Product model data (STEP).

A CAD system usually creates its own internal representation of the part and then converts it to IGES standard file formats for further activities. IGES describes points, line, curves, surfaces and other entities that can be seen in engineering drawings. IGES has been widely used in CAD/CAM environments.

It only defines each entity class and a sequence number in a neutral format file.

However, since the entity type can vary from one CAD system to another, as a result, there can be problems with file transfer.

### **2.5.7 User interface**

RE software systems need to be powerful, robust and user-friendly. In the future, it is desired that a RE system can automatically output some results after several interactions with the user. Of course the user can not be sure about the quality of these results, but it may help the user to fulfil his or her intentions. This means that users would be able to control the design process by altering the targets and not by altering the intermediate results. In addition, since targets are not uniquely defined, the system needs to satisfy different attempts from different users. A good way to achieve this would be the concept of modular tools to cope with the changing requests of the users (Podehl, 1996).

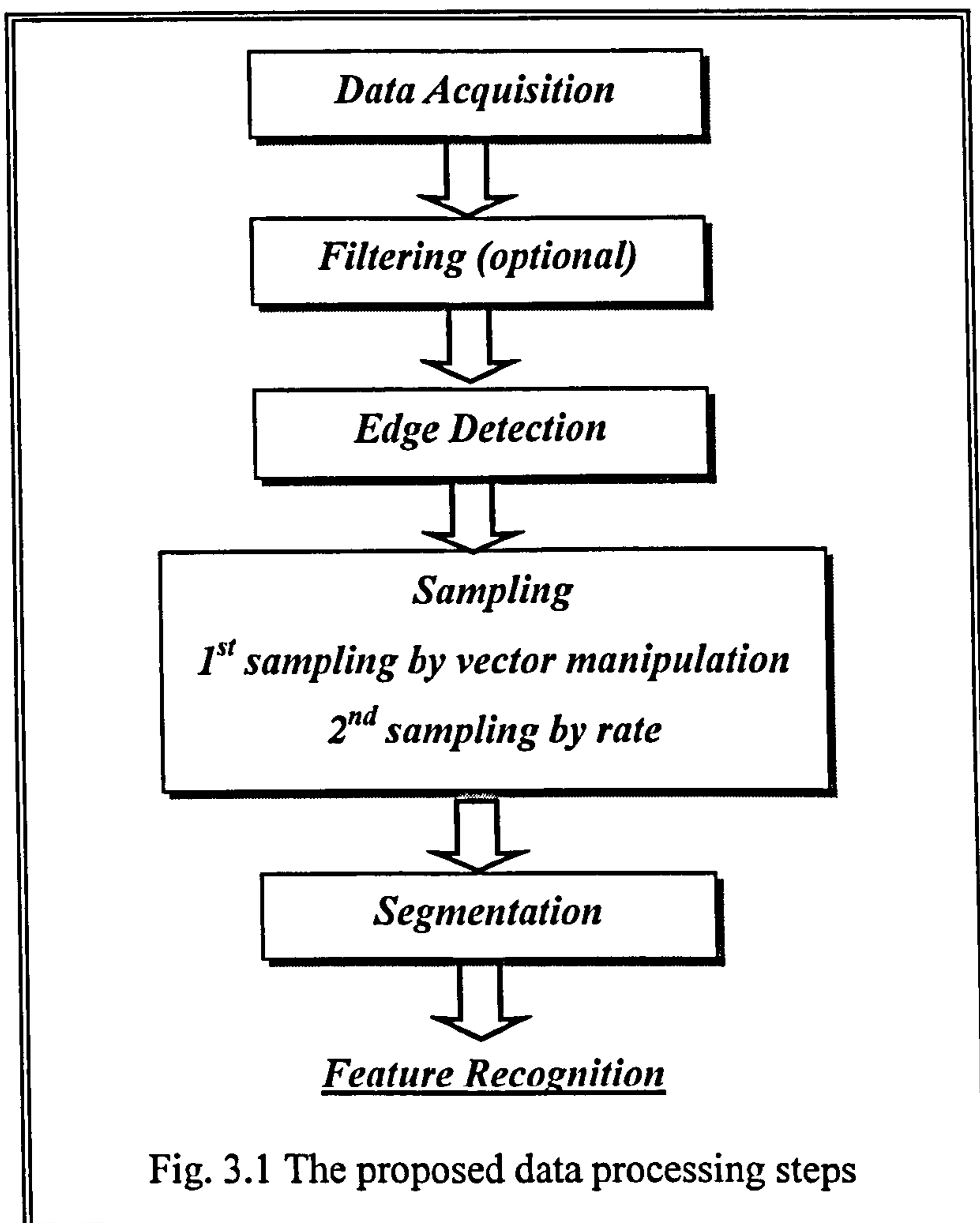


## **Chapter 3**

# **Scanned Data Processing and Edge Detection**

The main objective of RE is to convert a discrete scanned point set into topologically continuous CAD models. 3D scanners usually produce a large number of scanned points of varying size and accuracy. They are often termed “point clouds” due to their size and unformatted spatial arrangement. The initial scanned point data normally contain many redundant and erroneous points, which are not required for model reconstruction. For the system efficiency, it is desirable to appropriately reduce the data volume by removing measurement error and redundant points. A flat surface can be described by relatively less number of points. In contrast, high density of points is necessary to represent an area where curvature varies rapidly.

This chapter discusses point data processing algorithms such as filtering, sampling, edge detection and segmentation. The overall point data processing procedure is shown as flowchart (Fig. 3.1).



There are two processes to refine a point cloud set: *filtering* and *sampling*. Data filtering removes measurement noise, whilst data sampling is used to remove redundant points that can be easily seen in smooth surfaces of the part. System performance is critically dependent on these functions.

To reconstruct CAD models from a large amount of point data, it has to be manually split into sub-area associated according to constituent geometry such as surfaces or features. This crucial step is called '*segmentation*'. The segmentation is a manual selection process on a set of initial scanned point data

(Hafele, 1996). In this system, the user selects a proper area from the scanned points by defining four corners using a mouse 'window' function. The scanned point data in the 'window' selected are mapped to a surface or a feature. With the help of this 'window' segmentation, whole reconstruction processes regarding feature extraction can be divided into a set of segmented areas.

Since the selection of an appropriate area is an extremely difficult due to the complexity of the part, the segmentation process has been carried out by trial-and-error. In order to segment effectively, certain geometric characteristics of the initial data set are necessary because they can provide the user with guidelines for selection. In this research, "*edge points*" were chosen as the geometric characteristic, since they represent the boundary of constituent geometry. Trial-and-error in the segmentation process can be much reduced if the operator can reference edge points on the computer screen at the same time. Thus, if all the boundary edges of each region are known, it is relatively straightforward to reconstruct CAD models by mapping one region to a corresponding feature or surface. The dimensional parameters of a feature can be also determined from detected edge points. Hence, edge points carry geometric information which is crucial for effective model recognition and extraction in further stages of RE process.

There are two types of edge points in the scanned point data: sharp edges and smooth edges. They can be detected by investigating tangency discontinuities and the variation of curvature values, respectively. Any points where the direction of tangent vectors changes considerably may be a sharp edge, whilst it



is a smooth edge if corresponding curvature changes beyond a tolerance. Edge detection module was one of the most important prerequisites in this research, since the recognised features can be extracted from the information of detected edge points.

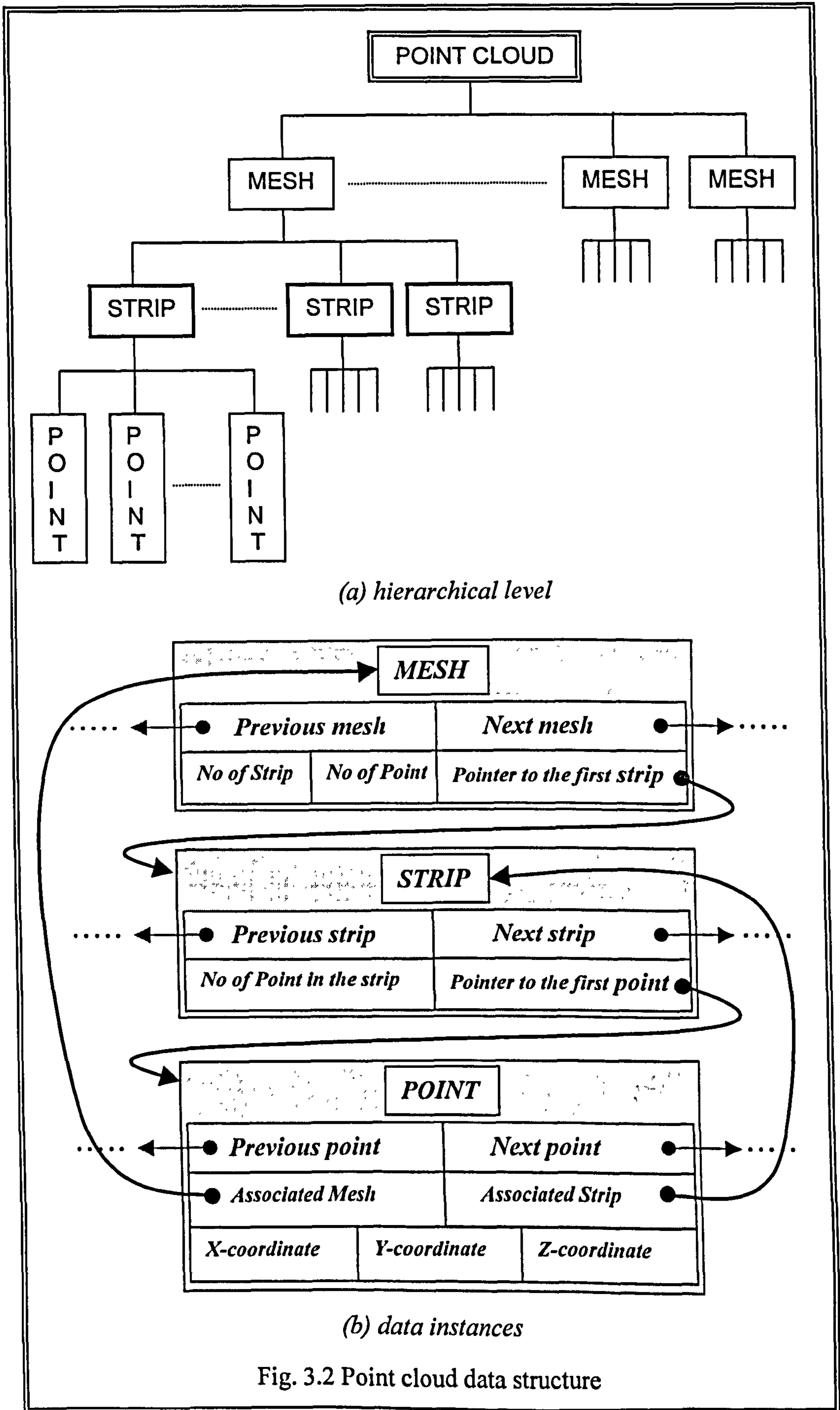
The data structure of point cloud must be compact and efficient to the extent that it helps the input scanned data to be successfully manipulated. The data structure of the ReTool system is first described in this chapter. Then the algorithms of filtering, edge detection, sampling and segmentation are discussed in depth.

### 3.1 Data structure for point clouds

In the design of a large application, data handling modules need to be created first from basic data types. Since the amount of scanned data is generally huge, the database is required to be implemented to handle the whole scanned data efficiently. Depending on the scanner and scanning strategy, different point formats can be acquired such as ASCII, IGES, and binary format, and, thus, the input point data has to be processed differently.

A Renishaw *Cyclone* is employed in this research to measure a target part, producing point cloud data, which as a whole define the shape of target surfaces. The ReTool system reads structured ASCII-formatted point data. Although ASCII point data just consists of three coordinates of scanned points ( $x, y, z$ ), the data structure developed is composed of four groups such as POINT CLOUD, MESH, STRIP, and its individual POINT for efficient handling.

Figure 3.2 (a) and (b) show hierarchical level of data groups and their instances, respectively.





As shown in Fig. 3.3, there are three types of scanned points: (1) a scanned data set with meshes, strips, and points; (2) a scanned data set with strips and points; (3) a scanned data set with points.

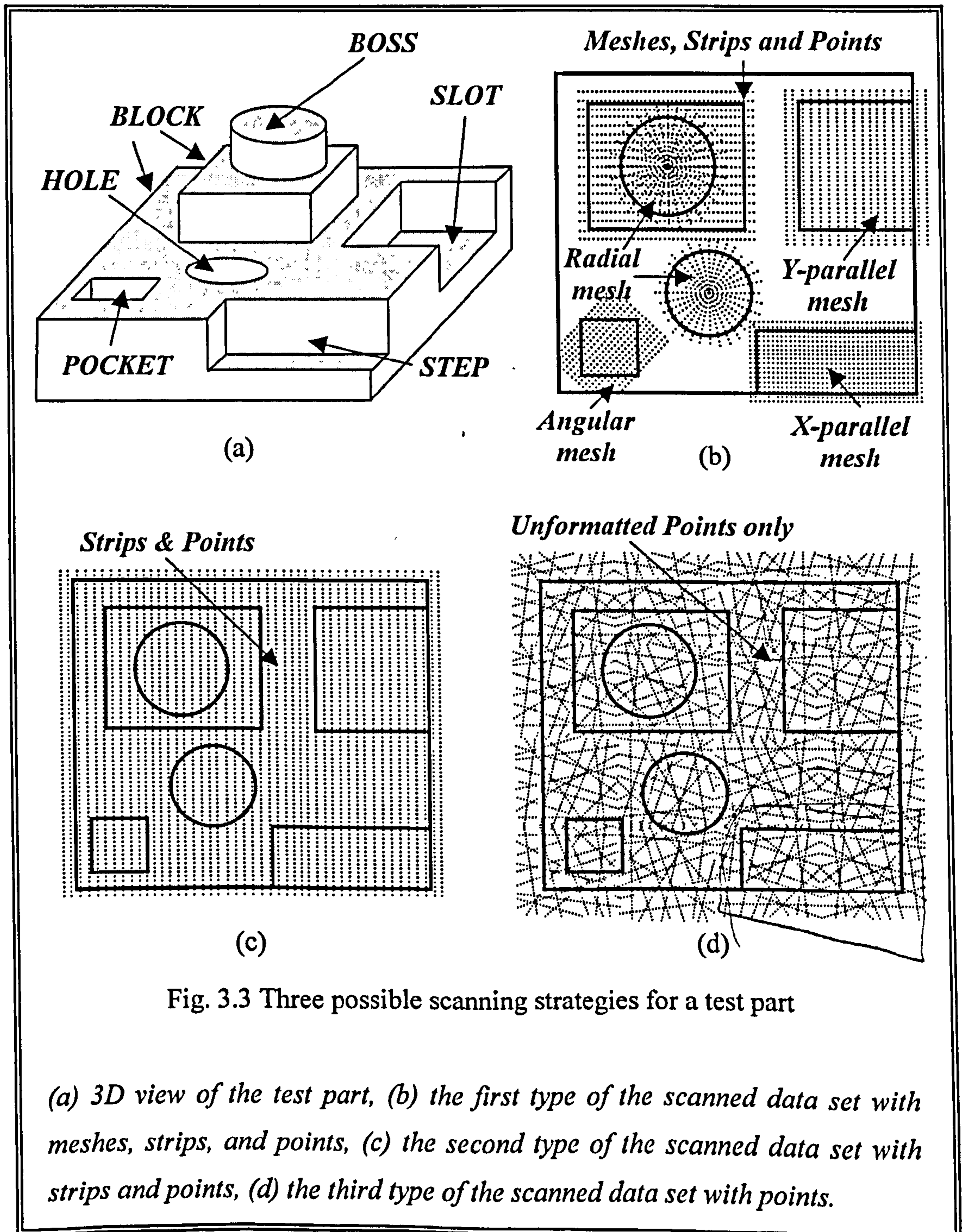


Fig. 3.3 Three possible scanning strategies for a test part

(a) 3D view of the test part, (b) the first type of the scanned data set with meshes, strips, and points, (c) the second type of the scanned data set with strips and points, (d) the third type of the scanned data set with points.

(1) A scanned data set consisting of meshes, strips, and points

This is an well-organised point set usually acquired by contact scanners that scan a part mesh by mesh, according to typical constitutional shapes/features. The scanned data might be like the pattern in Fig. 3.3 (b). The procedure for scanning data is to approach a feature at  $90^\circ$  to its boundary curves. *X-parallel*, *Y-parallel*, *angular* and *radial* meshes can be created depending upon the orientation of the feature. Angular meshes are composed of a series of parallel strip lines that are not parallel to the scanner axes. For instance, the *Cyclone* scanner (Renishaw, 1995) the grid definition dialogue is opened under the type “Angular” and then the required scanning angle is input under the heading “Angle”. All other parameters are as normal except that the point pitch and step-over will be measured at “Angle”. Radial scanning allows data capture to step around a circular form.

(2) A scanned data set being composed of strips and points

Since scanned point data using CMMs are on vertical planar surface section, all strips are usually parallel to an *arbitrary vertical plane* such as *x-z plane* or *y-z plane*. That is to say, the dimension of scanned points is reduced into 2D plane. Contact and some non-contact scanner can produce uni-directional point clouds. This is a common format as shown in Fig. 3.3 (c).

(3) A scanned data set consisting of unevenly spaced points

This data is often obtained by non-contact scanner (Fig. 3.3 (d)).



### 3.1.1 POINT CLOUD data structure

The header of a scanned point data set is stored and manipulated in a *POINT\_CLOUD* structure. In function calls involving putting and getting point clouds to and from the application database, the definitions the *typedef* mechanism of *C* is used to define the point data with the tag “*POINT\_CLOUD*” as new type:

```

typedef struct dbPointCloud {
    char    *ptcloud_name;
    long int NofPoints;
    long int NofStrips;
    long int NofMeshes;
    char    *ptcloud_type;
    union { // Pointer to the first mesh, strip and point
        struct dbMesh *start_mesh;
        struct dbStrips *start_strp;
        struct dbPoint *start_point;
    } start_entity;
} POINT_CLOUD;

```

*ptcloud\_name*: The name of the point cloud.

*NofPoints*: The number of points in the point cloud.

*NofStrips*: The number of strips in the point cloud.

*NofMeshes*: The number of meshes in the point cloud.

*ptcloud\_type*: The type of the point cloud. If *ptcloud\_type* is,

‘*MSP*’: The data consists of meshes, strips, and points.

‘*SP*’: The data consists of strips and points. Start entity is *start\_strp*.

‘*P*’: The data consists of points only. Start entity is *start\_point*.



### 3.1.2 MESH structure

As shown in Fig. 3.3 (b), every typical design feature can be separately scanned as a group. Therefore, the number of groups in the scanned data is generally the same to the number of individual features in the part. These groups are called *meshes*. A mesh defines the paths of the probe in a certain area of the part. *X*-direction, *y*-direction, *angular* and *radial* meshes can be created depending upon the features of the part. The correct procedure for capturing data is to approach a feature at  $90^\circ$  to its boundary edges. Once all of the required meshes have been planned, the data capture process can be executed. If the scanning strategy is such that each mesh data denotes just one design feature, it is straightforward to extract corresponding features in later stage. The standard C structure for mesh group is constructed as follows:

```

typedef struct dbMesh {
    long int NofPTinMesh;
    long int NofSTRP;
    struct dbStripsh *start_strip;
    struct dbMesh *next;
    struct dbMesh *prev;
} MESH;

```

***NofPTinMesh***: The number of points in the mesh. Must be greater than zero.

***NofSTRP*** : The number of strips in the mesh. Must be greater than zero.

***start\_strip*** : The pointer to the first strip.

***next*** : The pointer to next mesh.

***prev*** : The pointer to previous mesh.

### 3.1.3 STRIP structure

In the scanning of a part, as shown in Fig. 3.3 (b) and (c), scanned points are generally taken along a line aligned with one of the scanner axes by specifying the start and end points. For instance, the scanning probe moves *x*-direction in a 3D space or on a specific plane, and then increases a step size along the *y*-direction. These lines are called *strips*. Currently there are CMMs that are equipped with analogue scanning probes that can semi-automate the scanning process (Milroy, 1996). The strips are stored in an array whose element *STRIP* has the following structure definition:

```

.....
typedef struct dbStrips {
    long int NoPTinSTRP;
    struct dbPoint *start_pt;
    struct dbStrips *next;
    struct dbStrips *prev;
} STRIP;
.....

```

*NoPTinSTRP*: The number of points in the strip. Must be greater than zero.

*start\_pt* : The pointer to the first point.

*next* : The pointer to next strip

*prev* : The pointer to previous strip

### 3.1.4 POINT structure

The individual point data is stored and manipulated in a linked-list structure. Each element of the linked list is connected to its previous and next element. This data is a very low-level data, and does not have any geometric interpretations other than 3D coordinates. Each element contains its corresponding strip and mesh identification for efficient manipulations if the data set is acquired with strips or strips/meshes structure.

The point and its instances have the following structure definition:

```
typedef struct dbPoint {
    double *x, *y, *z;
    int AssociateSTRP;
    int AssociateMESH;
    struct dbPoint *next;
    struct dbPoint *prev;
}POINT;
```

*x, y, z:*                    *x, y, and z-coordinates of the individual point*

*AssociateSTRP:* Associated strip ID. *NULL* in the set containing points only.

*AssociateMESH:* Associated mesh ID. *NULL* in the set containing points only.

*next*                    :    The pointer to next point

*prev*                    :    The pointer to d previous point



### 3.2 Filtering Algorithms

A scanned data source is noisy if data derived from that source has a tendency to contain error. Filters can be normally used to reduce the error contained in a scanned data set. A *Median* or an *Average* filter algorithm is used to smooth a sorted point cloud in some commercial packages (*Surfacer*, 1995).

If the input data contains noisy points, the moving average filter with wide support ( $\Delta$ ) can be used (Gonzalez, 1992). Therefore, in this research, the filtering algorithm adopts a moving average filter. In its operation, the idea is to let  $\bar{P}_i$  be the outcome of averaging some values about  $\bar{P}_i$ .

For instance, a five-point moving average ( $\Delta = 5$ ) is as following,

$$\bar{P}_i = \frac{1}{5} (\bar{P}_{i-2} + \bar{P}_{i-1} + \bar{P}_i + \bar{P}_{i+1} + \bar{P}_{i+2}) \quad (3.1)$$

where,  $i = 3, 4, \dots, n-1, n-2$  ( $n$ : number of points)

This has the effect of smoothing out small deviations between scanned points, especially if the points are fairly close in value. In fact, practical filters typically have a local support, meaning that only a small number of nearby samples is used to define filtered results.

Generally a moving-average filter of support  $\Delta$  can be written as

$$f_{\text{avg}}[i] = \begin{cases} 1/\Delta & (\Delta: \text{The number of elements}) \quad \left( \begin{array}{l} \text{if } i \in \Delta \\ \text{otherwise} \end{array} \right) \\ 0 & \end{cases} \quad (3.2)$$

Filters can also be applied in the scanned points more than once. Some computer scientists call this *recursive filtering* (Gonzalez, 1992). The behaviour of filtered data is greatly affected by the support  $\Delta$ . A moving-average filter with wide support is desirable for noisy scanned points, however then the filtered output is slow to react to changes and consequently lags behind the original shape. Qualitatively, the wider the support  $\Delta$  or the more the number of iterations, which therefore blends more original data together, thus reducing the ‘sharpness’ of the raw data. Typically sharp edges disappear and are replaced by smooth blends, which in some cases may be acceptable, but in other cases may cause serious problems in detecting edges. Filtering with a small support  $\Delta$  more closely tracks an original scanned data set.

Figure 3.4 (a) shows an initial scanned data and its curvature values, and Fig. 3.4 (b) shows its filtered point data set. The point data set was filtered with five-point moving average ( $\Delta = 5$ ) with two iterations in that example.

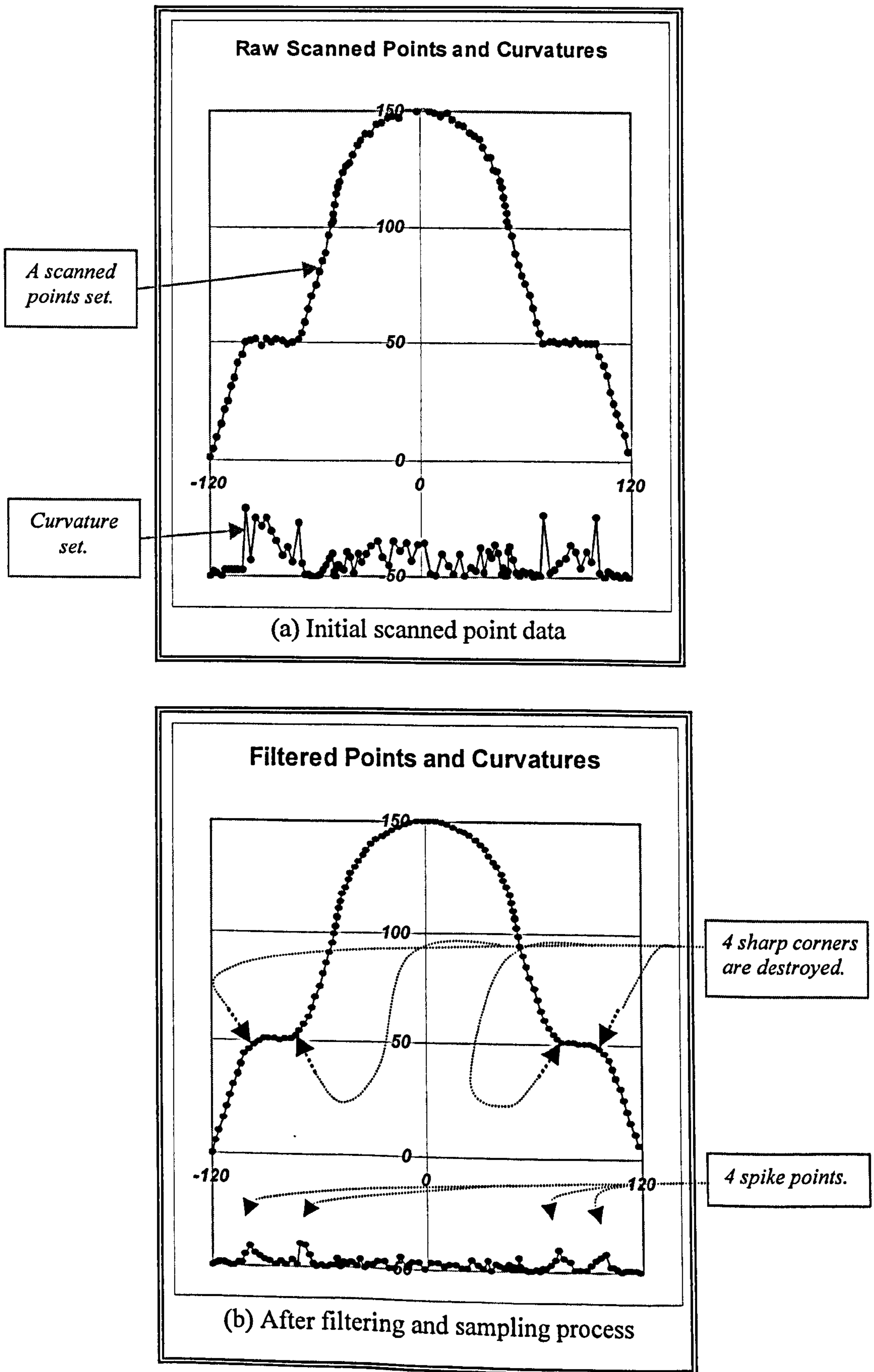


Fig. 3.4 Typical characteristics of scanned points and curvatures



After filtering, it is shown that the magnitude of curvature is reduced and the scanned point set is smoothed well. This is because all excessive error points are moved to the position which internal elastic energy is minimised.

If a high-accuracy contact scanner is used to scan the part, the uncertainty in different scanned points does not vary much. The assumption that a high-accuracy contact scanning devices do not generate bad data is reasonable since some scanners have the ability to neglect highly uncertain measurements (Besl, 1992). Therefore, for purposes of simplicity and relevance to scanning applications based on millions of scanned points, the removal of outliers was not implemented in the proposed application.

### 3.3 Edge Detection

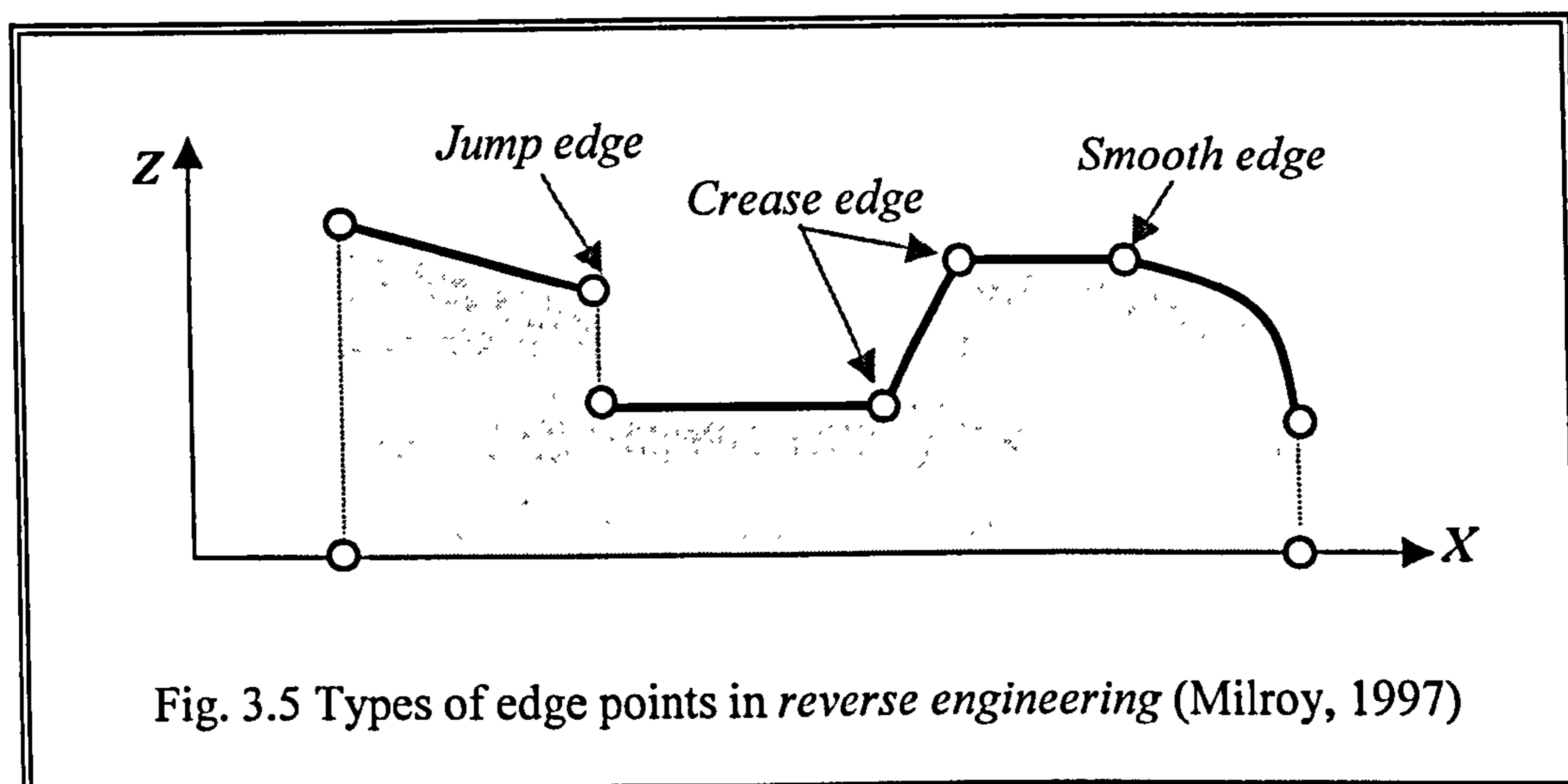
An edge in RE is either the outer boundary among constituent regions in a part with relatively distinct discontinuities, or internal boundaries in a region. In other words, edge points are those where a sharp or a smooth change occurs in the shape of region.

In this research, features are extracted from the edge points of a set of input data points. Therefore edges are geometric properties which are vital in this feature-based reverse engineering technology. However, the precise detection of the 3D edges is difficult on the computer screen, making this a laborious and error-prone process. Automation of this process is desired, but very difficult to implement (Milroy, 1997).

It is known that many research works has been investigated on just range image segmentation. The Laplacian Gaussian  $\nabla^2 G$  method has been used to detect boundary edges in the range image processing. In order to apply this method into a point cloud set, the z-coordinate of each scanned point needs to take the place of grey-level intensity at the pixels of an image. With checking the sign of the second derivative, all the edge points can be identified (Sarkar, 1991; Gonzalez, 1992). However, Chen Y. (1997a) described that it is often impossible to transform the reverse engineering data into a range image because not all scanned data are visible from any single view. Instead, they proposed a segmentation method based on B-spline approximation to scanned data.

Literature review to date shows that two basic edge-detection approaches have been investigated in this field, namely *edge-based* and *face-based* methods (Varady, 1997). The former works by trying to find boundaries in the point data representing edges between constituent shapes. The latter method reverse the logic and tries to find groups of points all having the same properties, and thus belonging to the same surface. After interpolating each surface, edges can be derived by intersection from the surfaces.

There are various types of edge point in RE as shown in Fig. 3.5.



It can be seen that any point where the direction of two tangent vectors estimated changes abruptly may be a sharp edge. Conversely a smooth edge is a point where the amount of curvature change is beyond a certain tolerance. The jump and crease edges can be detected by either the existence of tangency discontinuities or the variation of curvature values, whilst only the variation of curvature values can be used to detect the smooth edges.

If only sharp edges are only being sought, it is enough to find any place where the difference of two neighbouring tangent directions is suddenly changed. If

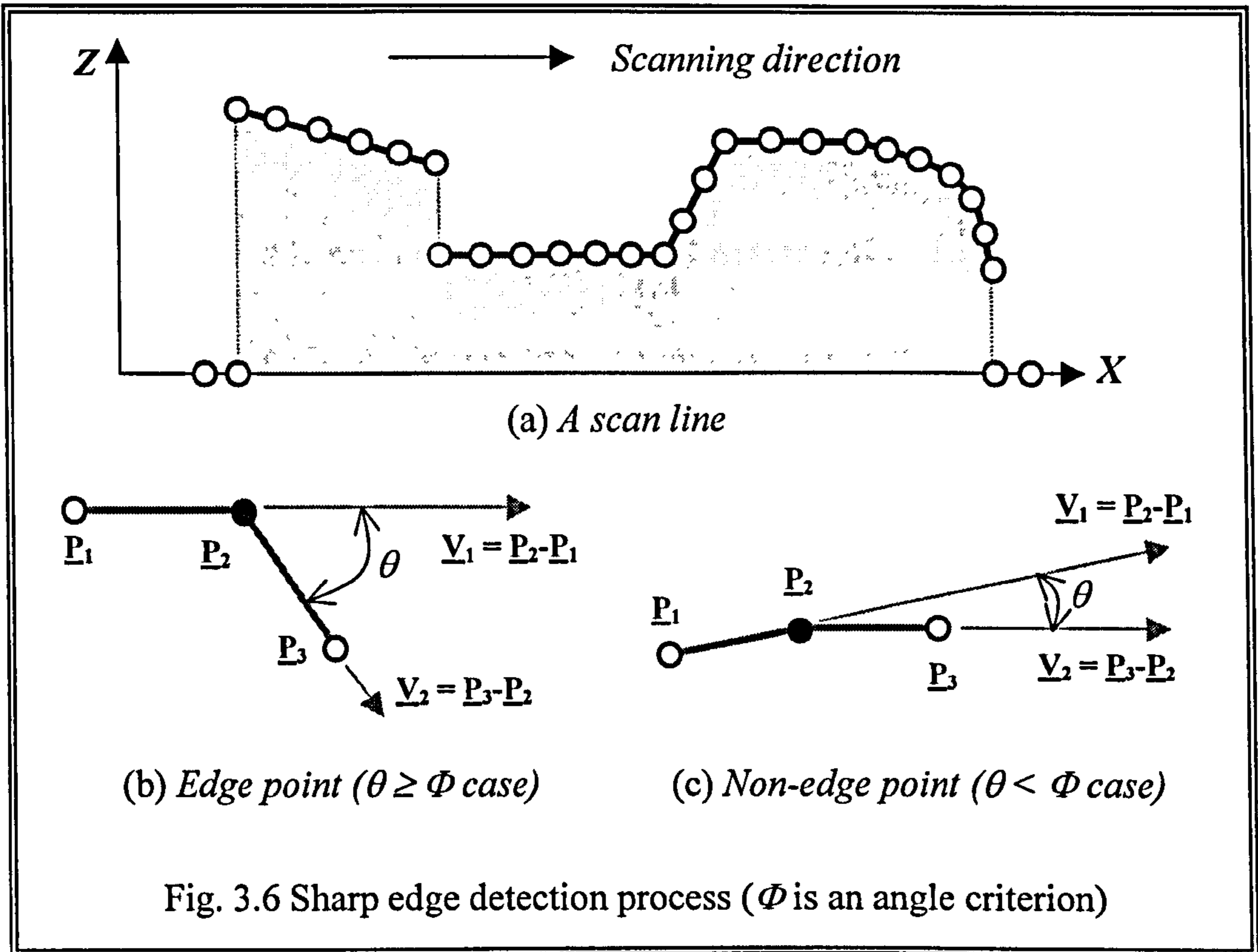


smooth edges are also being searched for, the places where curvature is not continuous along the scan line are required to be investigated. A whole set of consecutive curvature difference between two neighbouring points needs to be calculated first, and then sorted to obtain the maximum curvature difference, which may be a smooth edge.

In this research, a new edge-detection algorithm from a scanned point set is proposed, which is based on some simple arithmetic and derivative operators. The detailed algorithms are discussed in the following subsections.

### **3.3.1 Sharp edge detection by vector manipulations**

The sharp edge detection technique adopted in this research is computation of a local derivative operator. It is easily seen that the first derivatives (tangents) at each point along to a plane are invariable, and are only changed on curved surfaces. All the points at which tangency change value is greater than certain tolerance ( $\Phi$ ) are detected as sharp edge points because, in general, all boundary edge of prismatic feature are known to be sharp. The edge detection method involves interpreting the angle using three consecutive points in a strip line as shown in Fig. 3.6.



The method is based on a structured point cloud set. Figure 3.6 (a) shows a scan line that is parallel to  $X$ - $Z$  plane.  $\bar{P}_1$ ,  $\bar{P}_2$ , and  $\bar{P}_3$  are three consecutive points along to the scan line. The Euclidean unit tangent vector  $\bar{V}_1$  connecting two neighbouring points  $\bar{P}_1 = (x_1, y_1, z_1)$  and  $\bar{P}_2 = (x_2, y_2, z_2)$  is computed as follows:

$$\bar{V}_1 = \frac{\bar{P}_2 - \bar{P}_1}{\|\bar{P}_2 - \bar{P}_1\|} = \frac{(x_2 - x_1)\bar{i} + (y_2 - y_1)\bar{j} + (z_2 - z_1)\bar{k}}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}}$$

where  $\bar{i}$ ,  $\bar{j}$ ,  $\bar{k}$  are unit vectors to  $x$ ,  $y$ , and  $z$  axes, respectively.

Similarly,  $\bar{V}_2$  is,

$$\vec{V}_2 = \|\vec{P}_3 - \vec{P}_2\| = \frac{(x_3 - x_2)\vec{i} + (y_3 - y_2)\vec{j} + (z_3 - z_2)\vec{k}}{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2}}$$

Let us examine this process mathematically. Assume  $\Phi$  is the predefined angle criterion, and  $\theta$  is the angle between two unit vectors  $\vec{V}_1$  and  $\vec{V}_2$ . If  $\theta$  is greater than  $\Phi$  as shown in figure (b), the point  $\vec{P}_2$  may be an edge point. Otherwise, the point  $\vec{P}_2$  is a non-edge point as shown in figure (c). The next scanned point is used to calculate further angle vector in the same fashion through the series of scanned points. This process continues until all scanned data points are evaluated by the same process and a full set of edge points are detected. The value of predefined angle criterion  $\Phi$  needs to be carefully determined to the extent that the output tangent vectors are not sensitive to the measurement error. It is recommended that the value  $\Phi$  can be changed by the accuracy of utilised scanner.

### 3.3.2 Smooth edge detection

In order to detect smooth edge points, every curvature difference between two neighbouring points is computed. Then every point which curvature value changes dramatically is preserved for further utilisation.

As shown in Fig. 3.4(b), one characteristic provides a reasonable solution for detecting edges in a curvature plot: 'spike'. These spikes in the curvature plot give a clear indication of the location at which the points are to be partitioned into different sets. Hence, the presence of spikes can then be used to detect the



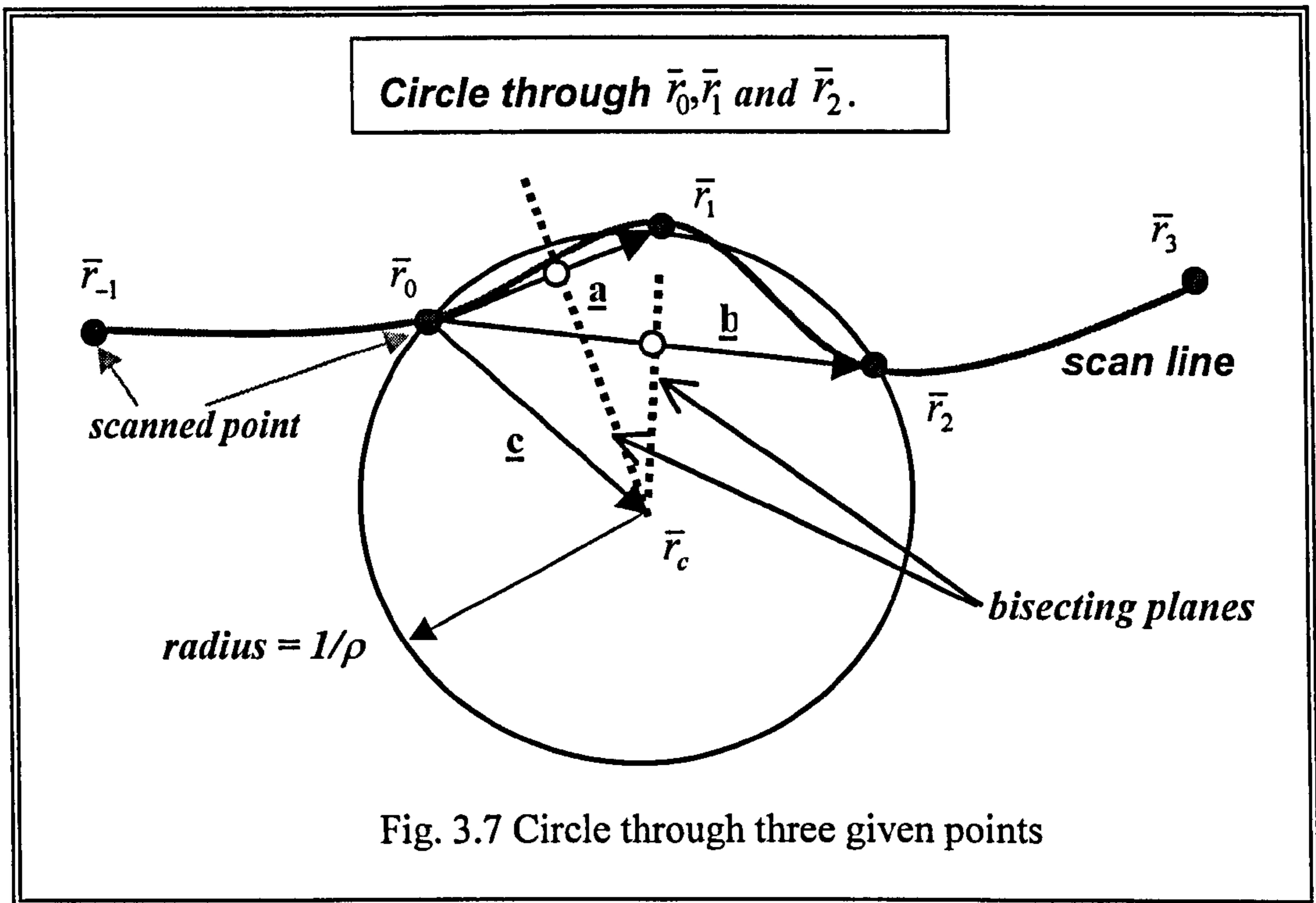
presence of smooth edges in the plot. Maximum curvature points are computed with equations in the section 3.3.2.1. This method is similar to the traditional method with range image data, but the z coordinate of each scanned point does not need to take the place of grey-level intensity since changes in geometry of a scan line can be computed by its curvature.

There are two mathematical algorithms to compute a curvature set along to a series of scanned point data for edge detection. One is by a circle which pass three consecutive points, and the other is by converting of scan data into spline curves. Once a series of curvature is computed, the associative smooth edge points can be detected.

### 3.3.2.1 Curvature calculation by a circle

Let the scanned points  $R_0$ ,  $R_1$ , and  $R_2$  have position vector  $\underline{r}_0, \underline{r}_1$  and  $\underline{r}_2$ . As shown in Fig. 3.7, then the centre of the circle through  $R_0$ ,  $R_1$ , and  $R_2$  is given by (Faux, 1981),

$$\underline{c} = \frac{b^2(a^2 - \underline{a} \cdot \underline{b})\underline{a} + a^2(b^2 - \underline{a} \cdot \underline{b})\underline{b}}{2|\underline{a} \times \underline{b}|^2} \quad (3.3)$$



And the radius is,

$$\text{Radius} = \frac{|\underline{a}||\underline{b}||\underline{a}-\underline{b}|}{2|\underline{a} \times \underline{b}|} \quad (3.4)$$

$$\text{where, } \underline{a} = \underline{r}_1 - \underline{r}_0,$$

$$\underline{b} = \underline{r}_2 - \underline{r}_0,$$

$$\underline{c} = \underline{r}_c - \underline{r}_0,$$

$$a = |\underline{a}| \text{ and } b = |\underline{b}|.$$

$$\text{or, } \text{Radius} = \sqrt{(x_c - r_{1x})^2 + (y_c - r_{1y})^2 + (z_c - r_{1z})^2} \quad (3.5)$$

$$\text{where, } \underline{r}_c = [x_c y_c z_c]^T : \text{centre of circle.}$$

Suppose we have a strip  $R$  of values,  $R_0, R_1, R_2, \dots, R_{n-1}, R_n$ . First three points ( $R_0, R_1, R_2$ ) in the strip are assigned to  $\underline{r}_0, \underline{r}_1$  and  $\underline{r}_2$ , so that the curvature of point  $R_1$  can be obtained as follows:

$$\text{Curvature} = \rho = 1/(\text{Radius from Eqn.(3.4)}) \quad (3.6)$$

Next three points ( $R_1, R_2$ , and  $R_3$ ) are also assigned to  $\underline{r}_0, \underline{r}_1$  and  $\underline{r}_2$ , then the curvature at point  $R_2$  can be computed. If the rest of points are input in the same fashion, the whole set of curvature value can be computed.

### 3.3.2.2 Curvature calculation by fitting a B-spline curves

In differential geometry, changes in the geometry of a curve can be measured by its curvature set. If scanned data of a strip line are fitted into a spline curve, all smooth edge points in the strip can be identified by checking the variation of curvature value.

Consider all scanned points of a scan line lie on a vertical plane. Then the B-spline fitting of these points is in fact a 2D problem. The procedure for fitting a B-spline curve is presented in this section. The B-spline curve is a special case of a NURBS curve<sup>1</sup>.

A NURBS curve of given degree  $d$  is well described by the following geometric handles (Piegl, 1995; Chen 1997a):

$$\text{Control vertices:} \quad V = \{V_0, V_1, \dots, V_n\}$$

$$\text{Weights:} \quad W = \{W_0, W_1, \dots, W_n\}$$

---

<sup>1</sup> If weights are identically one, NURBS curve will be effectively a tensor product B-spline curve.



Let  $V_i = \{x_i, y_i, z_i\}$ ,  $i = 0, 1, 2, \dots, n$ , be the  $i_{th}$  3D control point, then the  $i_{th}$  B-spline segment of degree  $d$  can be represented as:

$$r_i^d(u) = \frac{\sum_{j=i}^{i+d} w_j V_j N_{j-d,d+1}(u)}{\sum_{j=i}^{i+d} w_j N_{j-d,d+1}(u)} \quad (0 \leq u \leq 1, i = 0, 1, 2, \dots, n) \quad (3.7)$$

where  $N_{j-d,d+1}$  is the  $(j-d)_{th}$  B-spline blending function of degree  $d$ . The spline segments of B-spline curves meet with  $C^{d-1}$  continuity. In the case of a cubic B-spline ( $d = 3$ ), the segments will join with  $C^2$  continuity, e.g. the segments meet with identical first and second derivative at their junction points.

The choice of the parameterisation will have an effect on the fitted curve. Owing to the close spacing of the scanned points, the cumulative chord length provides a good approximation of the scanned point data. The computation of curvature values requires the calculation of the second derivative of the spline curve.

By evaluating Eq. (3.7) and simplifying subscripts  $i+1$ ,  $i+2$ ,  $i+3$  and  $i+4$  to  $1$ ,  $2$ ,  $3$  and  $4$ , respectively, the  $i_{th}$  segment of a cubic NURBS on the  $y$ - $z$  plane can be explicitly represented as:

$$y = \frac{a_0 + a_1 u + a_2 u^2 + a_3 u^3}{c_0 + c_1 u - c_2 u^2 + c_3 u^3} \quad (3.8)$$

$$z = \frac{b_0 + b_1 u + b_2 u^2 + b_3 u^3}{c_0 + c_1 u - c_2 u^2 + c_3 u^3} \quad (3.9)$$

$$\text{where, } \begin{cases} a_0 = w_1 y_1 + 4w_2 y_2 + w_3 y_3 \\ a_1 = -3(w_1 y_1 - w_3 y_3) \\ a_2 = 3(w_1 y_1 - 2w_2 y_2 + w_3 y_3) \\ a_3 = (w_1 y_1 + 3w_2 y_2 + 3w_3 y_3 + w_4 y_4) \\ b_0 = w_1 z_1 + 4w_2 z_2 + w_3 z_3 \\ b_1 = -3(w_1 z_1 - w_3 z_3) \\ b_2 = 3(w_1 z_1 - 2w_2 z_2 + w_3 z_3) \\ b_3 = (w_1 z_1 + 3w_2 z_2 + 3w_3 z_3 + w_4 z_4) \\ c_0 = -(w_1 + 4w_2 + w_3) \\ c_1 = 3(w_1 - w_3) \\ c_2 = 3(w_1 - 2w_2 + 3w_3) \\ c_3 = w_1 - 3w_2 - 3w_3 - w_4 \end{cases}$$

For parametric splines on the y-z plane, curvature can be calculated by the following equation (Mortenson, 1985):

$$k(u) = \frac{|y'z'' - z'y''|}{[(y')^2 + (z')^2]^{3/2}} \quad (3.10)$$

Now take the first and second derivatives of Eq. (3.7):

$$\dot{r}_i^d = \frac{d}{du} \left( \frac{R(u)}{h(u)} \right) = R(u) \left( \frac{-\dot{h}(u)}{h(u)^2} \right) + \frac{\dot{R}(u)}{h(u)} \quad (3.11)$$

$$\ddot{r}_i^d = R(u) \left( \frac{-\ddot{h}(u)}{\{h(u)\}^2} + \frac{2\{\dot{h}\{u\}\}^2}{\{h(u)\}^3} \right) + 2\dot{R}(u) \left( \frac{-\dot{h}(u)}{\{h(u)\}^2} \right) + \frac{\ddot{R}(u)}{h(u)} \quad (3.12)$$

where

$$R(u) = R_i^d(u) = \sum_{j=i}^{i+d} w_j V_j N_{j-d,d+1}(u)$$

$$h(u) = \sum_{j=i}^{i+d} w_j N_{j-d,d+1}(u)$$

By evaluating Eqs (3.11) and (3.12) with reference to Eqs (3.8) and (3.9),  $y', z', y''$  and  $z''$  can be calculated. Now substitute them into Eq. (3.10), curvatures can be obtained.

Clearly, the edge detection process strongly depends on the quality of the scanned data. In order to reduce the negative effect of excessive noise from this operator, it is necessary to employ adequate noise-filtering methods prior to edge detection since the calculation of tangent or curvature values is noise sensitive. In this research, fortunately, experiment results show that the scanner employed is reasonably accurate.



### 3.4 Sampling Algorithms

It is difficult to process a large number of data points in CAD systems. If the scanned part has large planar or smooth curved regions, it is desirable to remove some of the points in these regions within acceptable tolerance since huge point clouds can not be manipulated efficiently in the application. It needs to be determined precisely how many (or interval) points will be sampled. This amounts to saying that we can not reconstruct original parts correctly if the scanned data is sampled beyond the criterion. Unfortunately, the relationship between the number of scanned data points and the accuracy of a feature/surface fit is not straightforward to investigate, so that it remains a topic of current research (Thompson, 1995).

There is a variety of ways to reduce the number of scanned points for improving system memory and efficiency. Among them, three sampling methods are mainly employed (Sakar, 1991; Hosni, 1994); sampling by rate, sampling by vector manipulations, and sampling by voxels. All methods remove one of duplicated points located within given tolerance. In this application, first two sampling methods - sampling by rate and sampling by vector manipulations - were implemented. The edge points and two end points of each scan line need to be preserved regardless of applying a sampling process, because they are necessary to CAD model extraction.

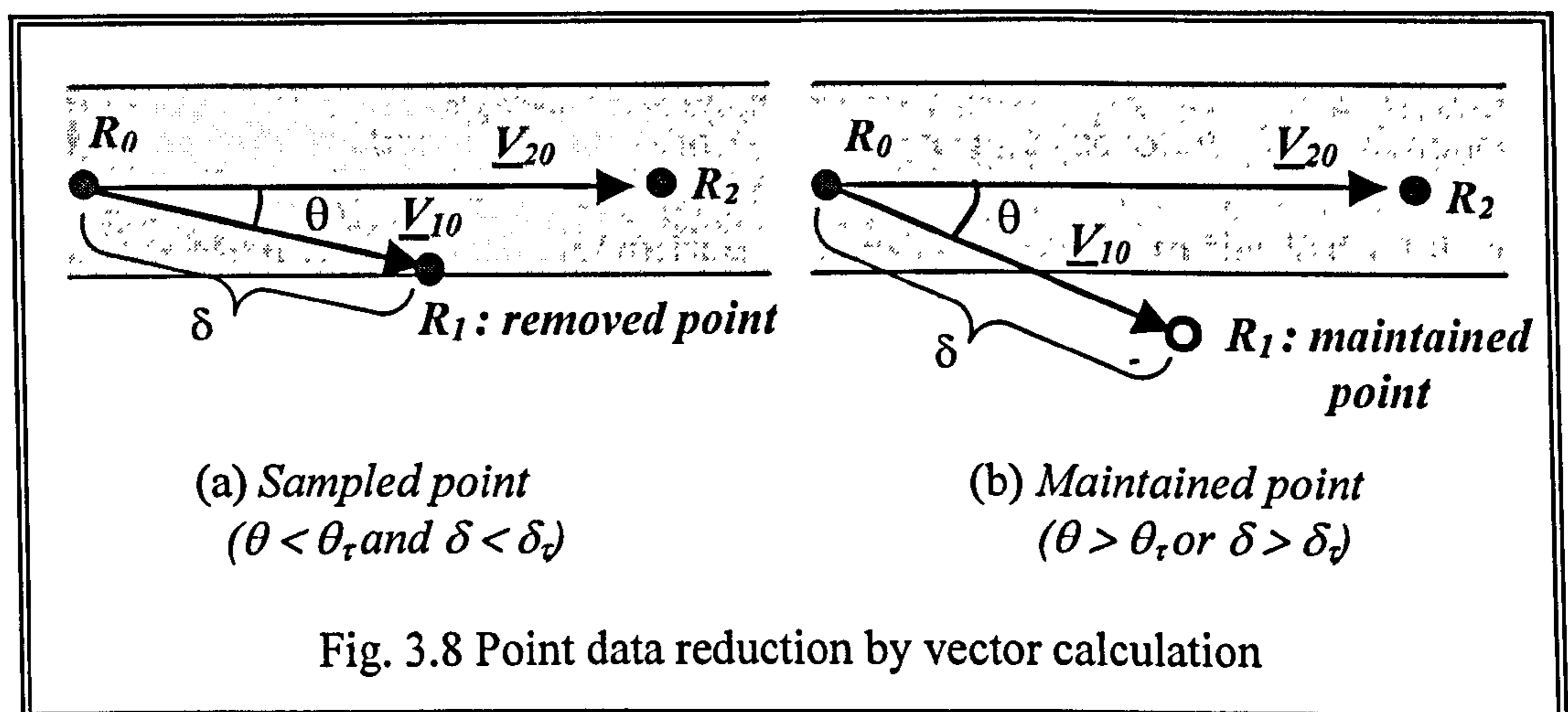
In this research, sampling by vector manipulations is first applied to initial raw scanned points. It reduces the number of points and preserves previously detected edge points. Then, sampling by rate is applied to the scanned data. Any

scanned point satisfied by at least one of the two sampling criterion discussed will be sampled. Each detailed sampling algorithm is discussed in the following subsections.

### 3.4.1 Sampling by vector manipulations

The sampling by vector manipulations can only be employed for regular and structured point clouds scanned by contact scanning devices. This method selectively removes points depending on the quality of the input point clouds, while retaining the information present in the data to a specified accuracy.

In this method, the current point ( $R_1$ ) and the next point ( $R_2$ ) are assigned into the module after previous point ( $R_0$ ) are selected as the reference points as shown in Fig. 3.8.



Then, two direction vectors are computed; one ( $\underline{V}_{10}$ ) is from  $R_0$  to  $R_1$  and the other ( $\underline{V}_{20}$ ) is between  $R_0$  and  $R_2$ . The idea is to remove the current point if the difference between two direction vectors is less than the predefined tolerance.

This process is quite similar to the sharp edge detection algorithm. The angle  $\theta$

between the two directional vectors  $\underline{V}_{10}$  and  $\underline{V}_{20}$  varies depending on the profiles of the strip line. The greater the change in the curvature of the curve between the two vectors, the greater the angle  $\theta$  between the directional vectors generated.

If the angle between two directional vectors ( $\theta$ ) is smaller than the predefined angle tolerance ( $\theta_t$ ), the current point ( $R_1$ ) is discarded from the strip line as shown in Fig. 3.8 (a). If the angle ( $\theta$ ) is larger than the tolerance,  $R_1$  should be maintained (Fig. 3.8 (b)). The next point data are used to compute the angle between two directional vectors in the same manner. The rest of the scanned points are determined to be removed or not by the angle of the directional vectors.

At the same time, another factor needs to be considered during sampling by vector manipulation. The distance between two considered points used in computing a directional vector must be taken into account. The distance guarantees that if the distance ( $\delta$ ) from a considered point to the last retained point is larger than the distance tolerance ( $\delta_t$ ), the considered point will be preserved. That is, if the distance of previous point and current point of a directional vector is larger than the distance tolerance, the current point is maintained even if it satisfies the angle criterion ( $\theta < \theta_t$ ). The point will be removed only when it satisfies two conditions simultaneously.

The data point is used to calculate directional vector and the distance in the same fashion, until all scanned data points are evaluated by the same process and a set of sampled points are obtained.



Finally, all scanned points require to be maintained in the region of the local curvature, whereas points lying in given tolerances are discarded in the region of smooth or straight part. This technique is very useful for sampling down dense data sets. The dialogue box of sampling process implemented and its whole flowchart is depicted in Fig. 3.9 and Fig. 3.10, respectively.

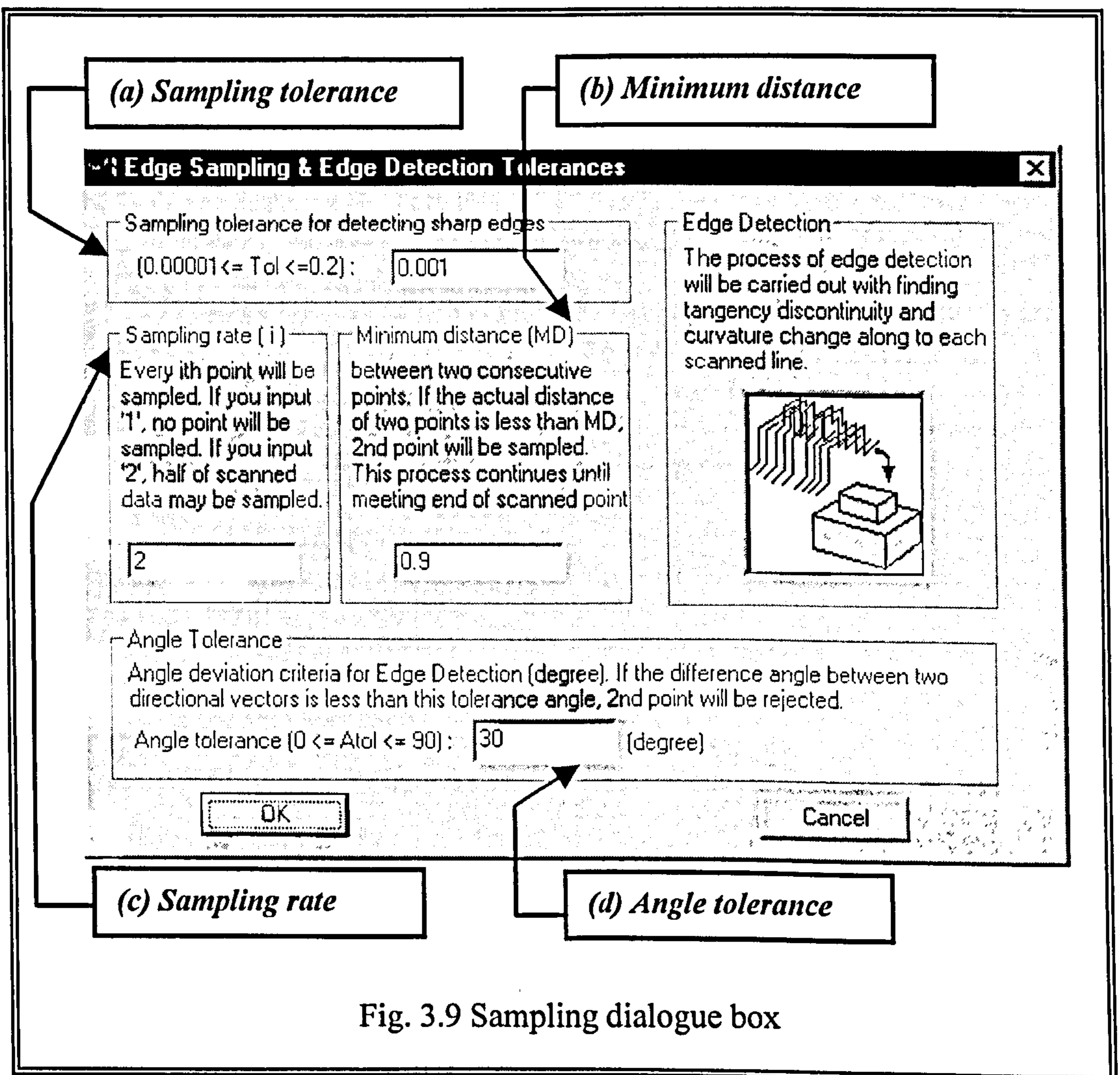


Fig. 3.9 Sampling dialogue box

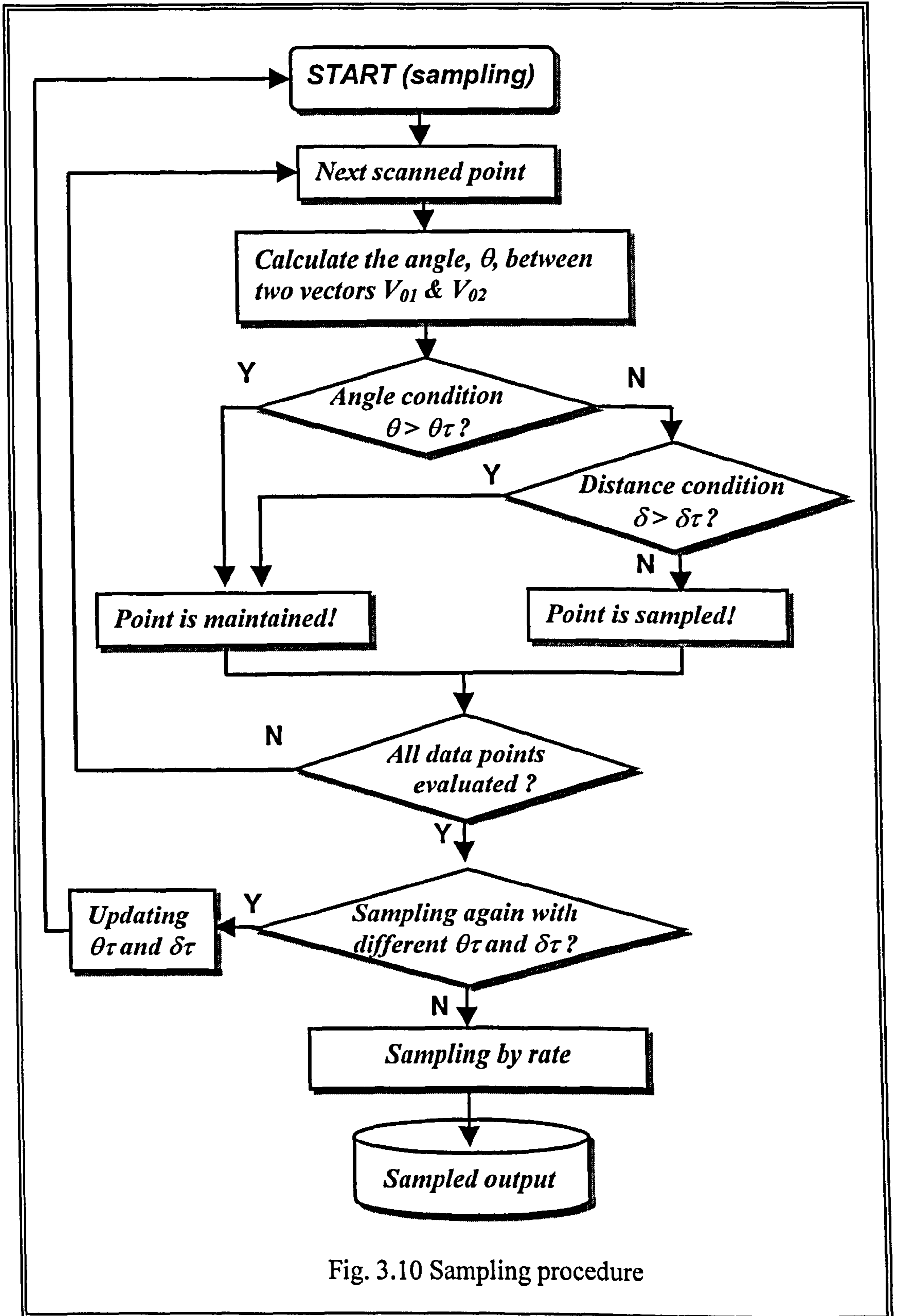
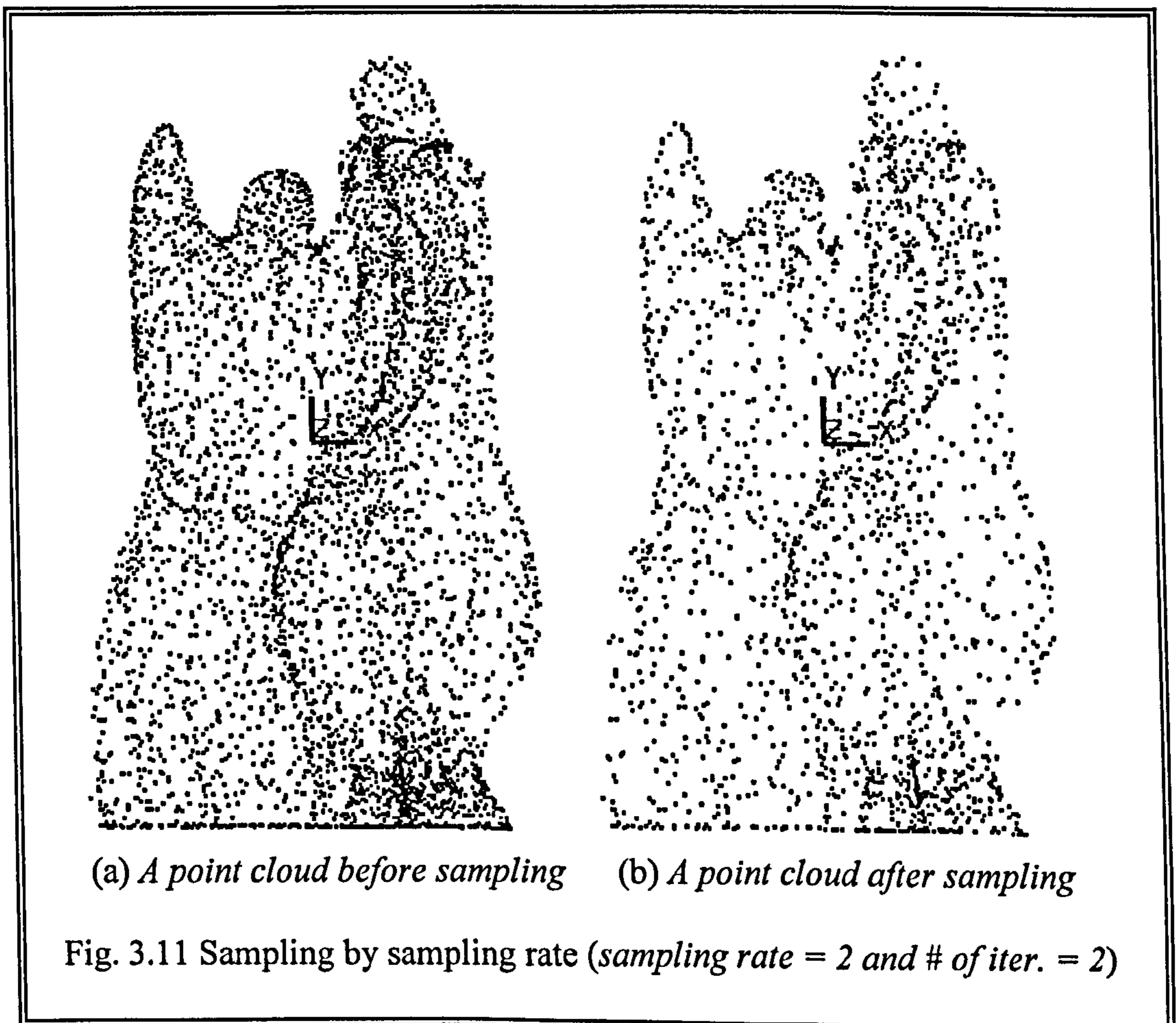


Fig. 3.10 Sampling procedure

### 3.4.2 Sampling by rate

Sampling by rate is a simple but powerful technique to reduce highly redundant points within a given tolerance. If the input point cloud is not arbitrary but structured, sample the point cloud every  $\alpha^{\text{th}}$  point in a scan lines and every  $\beta^{\text{th}}$  scan line in the total scan lines, where  $\alpha$  and  $\beta$  are users-input integer values. However, if the input point cloud is arbitrary or irregular, sample the point cloud every  $\alpha^{\text{th}}$  point only.

Figure 3.11 shows an example how much an input point clouds set can be reduced by this sampling process.





In this example, the sampling rate was 2 and the sampling process was applied two times. Therefore, almost half of the original point cloud was sampled since every point which sequence number was multiple of '2' along a scan line were removed except previously detected edge points. This method is particularly applicable to reducing the large number of point clouds acquired by laser and moiré scanner since they are often irregular and unstructured.

### 3.4.3 Sampling by voxels

Milroy M. (1996) proposed a sampling method that uses voxels. A rectangular volume that encloses the whole scanned data is subdivided into voxels. A voxel can be regarded as 3D pixel, so it has same size and is cube shaped. The total number of scanned data may be reduced by allowing just one point in a voxel. The number of scanned data in a voxel could be more than one point if the scanning strategy is so tight that the distance between one point and next one is less than the length of voxel. In this case, the point closest to the centre of voxel is maintained, and other points in the voxel are removed.

This method may be well applied to the unformatted and highly irregular scanned data acquired from non-contact scanners. Although this method requires relatively much time and high cost in memory, this algorithm eliminates huge amount of data point within given tolerance (the size of cube) in densely scanned area.

### 3.5 Data Segmentation

Segmentation is to logically divide the original point cloud set into regions, one for each surface/feature, so that each region contains just those points sampled from a particular surface/feature. Each of these regions is devoid of sharp edges and can be modelled with a single surface/feature. Fitting-surface or extracting-feature can be carried out only in the given region.

In fact, the segmentation is not a trivial problem. Since autonomous segmentation is one of the most difficult tasks in RE, it is usually carried out manually by the user. The solution is never unique since a designer's experience and his/her design knowledge are different. Segmentation must be terminated when the whole regions of the part have been isolated appropriately. The level of subdivision depends on the complexity of the part. This step determines the eventual success or failure of the RE process, so that effective segmentation leads to a successful CAD model reconstruction. Hence, segmentation is a crucial step towards the interpretation of discrete scanned data.

As the result of segmentations, the input point cloud set can be divided into several suitable distinctive prismatic features or free form surface regions. In the prismatic shaped regions, therefore, an appropriate quadric primitive (cylinder, block, or plane) may be fitted to give a more accurate model, and B-spline surfaces are fitted to the free form surface regions. They are precisely discussed in chapter 4 and 7.

## Chapter 4

# Surface Fitting to Scanned Data using *B-spline* Techniques

Whether it is cars, household equipment, or injection moulded plastic products, many of them have free form surfaces in their design. Thus, modelling of a sophisticated free form surface is increasingly necessary, because aesthetic shape is one of the key factors for designing value-added product (Sarkar, 1991; Lai, 1996). In the product design phase, CAD operators are often asked to create free form surfaces from a large number of scanned data points. A precise description technique is also required in RE to fit free form surfaces from point data.

In this work, every extracted feature is associated with the B-spline surfaces in the B-rep model of a part (i.e. a feature is bounded by a set of surfaces), so that B-spline surface techniques are also essential. Hence, this chapter deals with the techniques involved in generating a B-spline surface which interpolates a given set of scanned point data.



## 4.1 The Basic of Surface Modelling

The problem of fitting a free form surface to a set of point data has been widely investigated by many authors (Sarkar, 1991; Ma, 1995; Menq, 1996; Lai, 1996). Among the previous researches the most relevant to the current RE context is found in the work of Sarkar *et al* (1991). They proposed a new parameter optimisation method for reducing fitting error by the use of a nonlinear least-squares minimisation technique. The error between the scanned data points and the corresponding points on the fitted surface was minimised. More recently, Chen and Liu (1999) proposed a quadric surface extraction method based on genetic algorithms. They addressed that their method can be used to extract all types of quadric surfaces with a single surface representation. Chivate *et al* (1999) extended surfaces beyond their initial definition for constructing the B-rep model of the part using the tangent vectors at the edges.

The literature survey indicates that the most crucial current issue in surface reconstruction is the development of an appropriate surface optimisation scheme and its error analysis (Lin, 1998; Ueng, 1998).

### 4.1.1 The characteristics of surface modelling

Parts requiring free form surfaces in their design tend to fall into two main categories (Sarraga, 1984). One is a part which functions require free form surfaces such as car and aircraft bodies, impeller blades and the like. The other is a large group of parts whose functions do not require free form shapes, but which are designed with major free form features because of manufacturing

considerations and the aesthetic nature such as consumer electronics and household equipment.

The design of free form surfaces is a difficult process since it is labour intensive. The processing of point set utilising a commercial CAD tool is likely to be time consuming and doubts may occur about whether surface reconstruction is an adequate method for the creation of CAD models (Seiler, 1996). In addition to that, fitting surface models from a point cloud set is not only a task of high mathematical complexity, but also requires much creativity on the part of the user. It usually takes several iterations with testing of physical prototypes (Menq, 1996).

#### **4.1.2 Surface interpolation and approximation**

Free form surfaces are usually generated by interpolation or approximation methods (Piegl, 1995). In the interpolation method, a surface that passes through given points is constructed. It is applicable when the accuracy of scanned points is satisfied with predefined tolerance. It is desirable that these values be preserved by the fitted function. In approximation, however, surfaces do not necessarily pass through the given point data, but best represents them in an average manner. This is also termed “data smoothing”, and it is used when the scanned data may be subject to high scanning errors. If scanned point data contain huge noise errors, approximation is more suitable than interpolation for overcoming the noise errors and reducing computational burden. In that case, it is important for the surface to capture the “shape” of the data, but not to “wobble” its way through every point (Piegl, 1995). To minimise the approximation error,

the least square approximation which minimises the sum of the squares of the error is the most common method (Sarkar, 1991; Lai, 1996).

### 4.1.3 Surface modelling techniques

As discussed in chapter 2, several parametric polynomial surface models and boundary interpolating surface models are most popularly implemented in current commercial CAD packages. Commercial surfacing softwares, including Imageware *Surfacer*<sup>TM</sup>, Parametric *Pro/SCAN-TOOLS*<sup>TM</sup>, Cyberware *Cyserf*<sup>TM</sup>, Geomagic *Shape*<sup>TM</sup>, or Geomagic *Wrap*<sup>TM</sup>, have recently become available. These tools fit spline patches to scanned data points. The detailed commercial packages for RE are listed in *Appendix 2*.

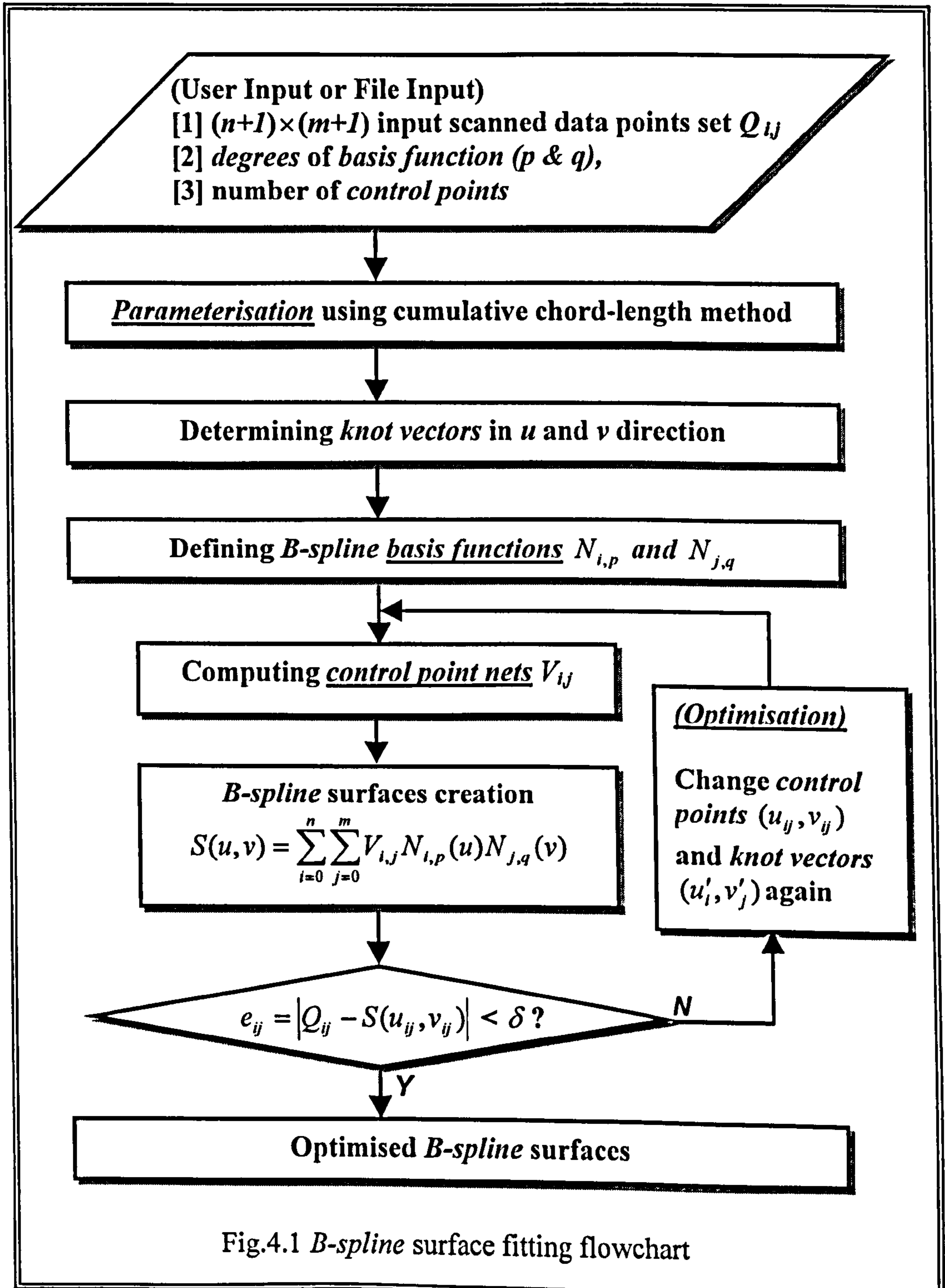
Typical examples of parametric polynomial surface models encompass the Ferguson surface, Bezier surface, B-spline surface and non-uniform rational B-spline surface (NURBS). In comparison, coons surface, ruled surface, and sweep surface are examples of boundary interpolating surface models, which are widely applied in the aircraft or shipbuilding industries (Menq, 1996).

Among the parametric polynomial surface models, B-spline techniques are one of the most widely accepted modelling functions in an interactive computer environment because of their ease of representation (Blanc, 1996). Once B-spline surfaces representing a physical model are generated, the design is completed with the production of specific features.



## 4.2 B-spline Surfaces Fitting Algorithm in this Research

Figure 4.1 shows the flowchart implemented for B-spline surface fitting to scanned points.



As shown in the figure, this algorithm is divided into four major steps: (1) parameterisation, (2) calculation of the basis function, (3) evaluation of control point net and (4) optimisation of two parameter values,  $u$  and  $v$ .

The process starts by establishing mesh points from the input point data. It next determines orders of basis function and number of control points. According to the requested tolerance value, it uses chord-lengths of input scanned point data for parameterisation. Next, knot vectors are determined, and then control points are calculated using the *Gauss-Jordan elimination* (Press, 1992; Lai, 1996). Once all the parameter values are calculated, B-spline surface can be created. The fitting error between the scanned point data and generated B-spline surfaces can be computed. If the error is beyond the pre-defined tolerance, the B-spline surface needs to be recreated using modified control points and knot vectors until the fitting error is less than the tolerance value. Detailed algorithms of individual step are discussed in the following section.

### 4.3 Fitting a B-spline surface to a set of data points

Given a set of scanned data for a single free form surface, a B-spline surface is fitted here to represent the original of the part. The basic theories of the B-spline techniques implemented in ReTool system are now described.

#### 4.3.1 The definition of B-spline basis functions

A Renishaw CMM (1998) employed in this research to capture point data with rectangular grid lines.

Suppose a given set of  $(n+1) \times (m+1)$  weighted scanned data points  $Q_{ij}$ , where  $i = 0, \dots, n$  and  $j = 0, \dots, m$ . In order to fit a B-spline surface from these  $(n+1) \times (m+1)$  data points,  $(n+3) \times (m+3)$  control points  $\{V_{ij}: i=0, 1, \dots, m+2; j=0, 1, \dots, n+2\}$  need to be computed. Since  $(n+1) \times (m+1)$  are given whilst  $(n+3) \times (m+3)$  are unknown,  $2m+2n+8$  of additional boundary conditions have to be determined in order to resolve all control points. These boundary conditions can be assigned as (Lin, 1998):

- $\{S_{0j}, S_{mj}: j=0, 1, \dots, n\}$ : boundary tangents in  $u$ -dir.; numbered  $2n+2$  in total.
- $\{t_{i0}, t_{in}: i=0, 1, \dots, m\}$ : boundary tangents in  $v$ -dir.; numbered  $2m+2$  in total.
- $\{x_{00}, x_{m0}, x_{0n}, x_{mn}\}$ : 4 twist vectors.

A degree  $(p, q)$  B-spline surface with  $n$  and  $m$  control points in the  $u$  and  $v$  parametric directions is mathematically defined as (Piegl, 1995),



$$Q \approx S(u, v) = \sum_{i=0}^n \sum_{j=0}^m V_{i,j} N_{i,p}(u) N_{j,q}(v) \quad (4.1)$$

where  $S(u, v)$  is the position vector along the surface as a function of the parameters  $u$  and  $v$ ;  $V_{i,j}$  are the  $(n+1)(m+1)$  unknown control point net for the patch;  $N_{i,p}(u)$  and  $N_{j,q}(v)$  are the B-splines basis functions of degree  $p$  and  $q$  in the  $u$  and  $v$  directions defined over suitably chosen knot vectors.

The degree of polynomial does not exceed  $p$  inside each  $[u_i, u_{i+1})$  interval and the position and  $[1 \text{ to } (\text{degree}-1)]$  derivatives are continuous.  $p$  and  $q$  are independent of  $m, n$  as long as  $m - p + 2 > 0, n - q + 2 > 0$ . A B-spline surface can express continuous patches by multiplying the knot vector set in each parameter direction.

The  $i_{th}$  normalised B-spline basis functions of degree  $p$  and  $q$  are defined by the *Cox-deBoor recursion formulas* as follows,

$$N_{i,0}(u) = \begin{cases} 1 & \text{for } u'_i \leq u < u'_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u'_i}{u'_{i+p} - u'_i} N_{i,p-1}(u) + \frac{u'_{i+p+1} - u}{u'_{i+p+1} - u'_{i+1}} N_{i+1,p-1}(u) \quad (4.2)$$

$$N_{j,0}(v) = \begin{cases} 1 & \text{for } v'_j \leq v < v'_{j+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{j,p}(v) = \frac{v - v'_j}{v'_{j+q} - v'_j} N_{j,q-1}(v) + \frac{v'_{j+q+1} - v}{v'_{j+q+1} - v'_{j+1}} N_{j+1,q-1}(v) \quad (4.3)$$

The *knot vectors*,  $U = [u'_0, u'_1, \dots, u'_r]$  and  $V = [v'_0, v'_1, \dots, v'_s]$ , satisfy with  $u'_i \leq u'_{i+1}$  and  $v'_j \leq v'_{j+1}$  condition respectively, which means that the elements of the knot vector have a non-decreasing sequence of real numbers. The  $u'_i$  and  $v'_j$   $\{i=0, 1, \dots, r; j=0, 1, \dots, s\}$ , *knots*, are elements of the knot vectors along  $u, v$  parametric direction.

$N_{i,0}(u)$  is a step function, equal to zero everywhere except on the half-open interval  $u \in [u'_i, u'_{i+1})$ , which is called  $i_{th}$  knot span. From the equation (4.2), it can be seen that a B-spline of degree  $p$  in the  $i_{th}$  span is the weighted average of the B-splines of degree  $p-1$  in the  $i_{th}$  and in the  $(i+1)_{th}$  span. Computation of a set of basis functions requires specification of a knot vector  $U$  and  $V$ , and the degree  $p$  and  $q$ .  $N_{i,p}(u)$  are piecewise polynomials, defined on the entire real line; generally only the interval  $[u'_0, u'_m]$  is of interest. The parametric values are computed using the chord length parameterisation method and using these values, the knot vectors are calculated using the averaging technique (Piegl, 1995).

### 4.3.2 Parameterisation

Surface parameterisation is an essential step influencing final generated shape and its topology. It gives parameter values to the scanned data points in each segmented region. Parameterisation of scanned data is usually achieved by three methods (Ma, 1995; Sinha, 1996): *uniform parameters*, *cumulative chord-length parameters* and *centripetal parameter*. These methods are based on the

assumption that the points are arranged in a grid and are therefore not suitable for randomly scattered scanned points.

Uniform parameterisation is the simplest way to assign parameters to scanned data. This method is sometimes unsatisfactory because the distribution of the scanned points is not taken into account. In this case,

$$u_k = \frac{k-1}{n-1} \quad 1 \leq k \leq n$$

$$v_k = \frac{k-1}{m-1} \quad 1 \leq k \leq m$$

Chord-length parameterisation employed in this research is to assign  $u$  and  $v$  values corresponding to each scanned point along to a strip line. The same procedure is repeated for every strip defined by  $u = 0$  and  $1$  to assign the  $v$  values. This is the most widely used method, and it is generally adequate (Ma, 1995; Piegl, 1995).

If a set of scanned points  $Q_k$  is given, the parameter value  $u_k$  can be computed as,

$$d = \sum_{k=1}^n |Q_k - Q_{k-1}| \quad : \text{total chord length}$$

Then,  $u_0 = 0,$

$$u_k = u_{k-1} + \frac{|Q_k - Q_{k-1}|}{d} \quad (\text{where } k=1, \dots, n-1) \quad (4.4)$$

and

$$v_0 = 0,$$



$$v_k = v_{k-1} + \frac{|Q_k - Q_{k-1}|}{d} \quad (\text{where } k=1, \dots, m-1)$$

This is in fact a scaled approximation of the arc length parameterisation, so the distribution of  $u$  and  $v$  tends to be proportional to the distance between scanned points.

Centripetal parameterisation, proposed by Lee, E. (1989), gives better results when the data points take very sharp turns (Piegl, 1995).

$$d = \sum_{k=1}^n \sqrt{|Q_k - Q_{k-1}|}$$

Then,  $u_0 = 0, u_1 = 1$

$$u_k = u_{k-1} + \frac{\sqrt{|Q_k - Q_{k-1}|}}{d} \quad (4.5)$$

### 4.3.3 Selection of knots

The designer chooses the number of knots, or, in other words, the number of segments. The number of segments clearly depends on the number of scanned data points and the predefined tolerance value. The knots control the segmentation of the surface - surface segments always start and end at knot values (Parasolid, 1998). As the number of knots increases, the sum of the squares of the individual errors decreases.

Three types of knot vectors are generally used, i.e. *non-uniform*, *uniform* and *open knot vectors*. The choice of knot vectors has a significant influence on the fitted surface. However, the determination of knot values is not easy to automate

(Ueng, 1998). Non-uniform knot vectors may have either unequally spaced or multiple internal knot values, and are suitable to precisely describe surfaces having sharp corners. In a uniform knot vector, individual knot values are evenly spaced, so that  $(u'_i - u'_{i-1}) = a$  for all intervals, and  $a$  is a real number. In most practical applications, the knot sequence starts at zero and often is normalised in the range  $[0 \text{ to } 1]$ . An open knot vector has a multiplicity of knot values at the ends equal to the order  $p$  of the basis function, whilst the internal knot values are evenly spaced. The multiplicity of a knot affects the parametric continuity at this knot. The open knot vector was employed in this research since it provides a realistic description of the shape of the boundary (Lai, 1996; Ueng, 1998).

The open knot vector  $U$  and  $V$  is formally given by

$$\begin{aligned}
 U &= \left\{ \underbrace{0, \dots, 0}_{p+1}, u'_{p+1}, \dots, u'_{r-p-1}, \underbrace{1, \dots, 1}_{p+1} \right\} \\
 V &= \left\{ \underbrace{0, \dots, 0}_{q+1}, v'_{q+1}, \dots, v'_{s-q-1}, \underbrace{1, \dots, 1}_{q+1} \right\}
 \end{aligned} \tag{4.6}$$

where,  $U$  and  $V$  have  $r+1$  and  $s+1$  knots, respectively.

In the open knot vector method, the knots reflect the distribution of the parameter  $u_n$ . The knot-value sequence means that the basis function are no longer the same for each interval, but rather vary from patch to patch. The relation between the number of knots ( $r+1$ ), the degree ( $p$ ) of  $N_{i,p}$  and the number of control points ( $n+1$ ) is given by  $r = n + p + 1$ . The surface always interpolates the first and last control point set. The Bezier surface is a special case of an open uniform B-

spline surface, where the number of control points used equals the order of the surface.

#### 4.3.4 Insertion of points

Although the scanned points are uniformly planned on the  $x$  &  $y$ -plane, they may be irregularly distributed on the part surface. In order to fit a set of unevenly spaced scanned points, it is often necessary to insert points into the input point data to eliminate oscillation on the surface created by making the number of data points are the same among scan lines. However, due to the insertion of point, the overall amount of point data can be increased considerably.

Lin *et al* (1997) proposed an equation to insert points on the scan line as follows. Consider that  $Q_s$  (start) and  $Q_e$  (end) are two adjacent scanned points,  $Q_m$  (middle) denotes the  $m$ <sub>th</sub> interpolated point between  $Q_s$  and  $Q_e$  and  $d_{min}$  represents the allowable distance between two adjacent points. Then,  $Q_m$  can be determined by means of the following equations:

$$\begin{cases} Q_m = Q_s + \frac{i}{n}(Q_e - Q_s) \\ n = \frac{|Q_e - Q_s|}{d_{min}} \end{cases} \quad (n \text{ is an integer}) \quad (4.7)$$

where  $n$  denotes the number of interpolated points between  $Q_s$  and  $Q_e$ .

By this method, the distribution of point data on the part surface becomes more uniform and the unexpected oscillation problem on the fitted surface can be



eliminated. The allowable distance  $d_{min}$  determines the performance of the interpolation algorithm. Herein  $d_{min}$  is chosen as the root-mean-square of the length of the line segments on one surface.

#### 4.3.5 Surface optimisation

The error analysis between an original point set and its fitted surface is a crucial step. Lin, A. *et al* (1998) proposed a methodology for resolving the coordinate alignment problem of machined parts and their original products. In their work, a least-squares method was adopted to search for a coordinate transformation matrix, and then the transformed points were compared with the fitted surface of a machined workpiece. However their proposed method has a problem to compare reference points with products which have a region of steep-slope. This is because the points were referenced along to a Cartesian coordinate, not to actual surface of the product. The error margin can be beyond the tolerance even if the referenced points satisfy the condition.

In this research, the surface-fitting algorithm was formulated as the minimisation of an error expression. That is, parameter  $u$  and  $v$  were continuously updated until the error between fitted surface and original point set was less than predefined tolerance.

Suppose again a given set of  $(n+1) \times (m+1)$  weighted data points  $Q_{ij}$  where  $i = 0, \dots, n$  and  $j = 0, \dots, m$ . The parameters  $u_{ij}$  and  $v_{ij}$  are optimised by the following error expression (Lai, 1996; Ueng, 1998):

$$e_{ij} = |Q_{ij} - S(u_{ij}, v_{ij})| \quad (4.8)$$

where  $Q_{ij}$  represents the scanned point,  $S(u_{ij}, v_{ij})$  denotes the corresponding point on the fitted surface, and  $e_{ij}$  represents the minimum distance of the scanned point  $Q_{ij}$  to the fitted surface. The deviation analysis can also be performed in  $x$ ,  $y$ , and  $z$  components as follows:

$$\begin{aligned} e_{ij,x} &= |Q_{ij,x} - S_x(u_{ij}, v_{ij})| \\ e_{ij,y} &= |Q_{ij,y} - S_y(u_{ij}, v_{ij})| \\ e_{ij,z} &= |Q_{ij,z} - S_z(u_{ij}, v_{ij})| \end{aligned} \quad (4.9)$$

in which  $e_{ij,x}$ ,  $e_{ij,y}$  and  $e_{ij,z}$  are deviation of  $x$ ,  $y$ , and  $z$  components, respectively, at the  $i$  &  $j^{\text{th}}$  point. For any point, the error is:

$$E_{ij} = \sqrt{e_{ij,x}^2 + e_{ij,y}^2 + e_{ij,z}^2} \quad (4.10)$$

The average error is defined as:

$$E_{average} = \frac{1}{(m+1)(n+1)} \sum_{i=0}^n \sum_{j=0}^m E_{ij} \quad (4.11)$$

If  $E_{average}$  is less than a predefined tolerance value, the fitted surface is accepted.

Otherwise, parameter  $u$  and  $v$  are updated and surfaces are regenerated until satisfactory.

### 4.3.6 Advantages of B-spline representation

B-spline fitting techniques can well approximate free form shapes while maintaining  $C^2$  continuity. The following properties are useful and crucial advantages in RE.

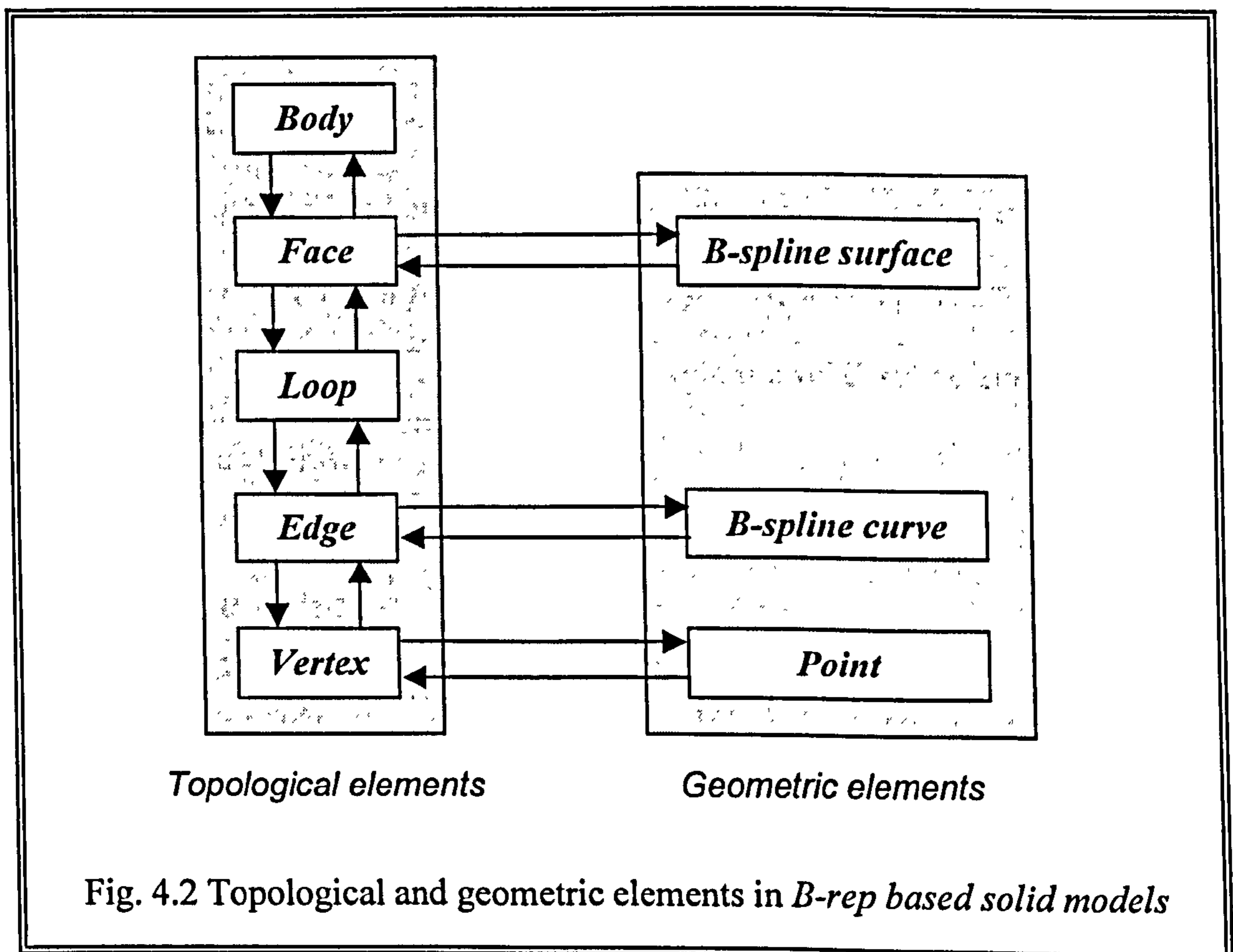
- A  $d_{\text{th}}$  degree B-spline is  $d-1$  times continuously differentiable (Sinha, 1996).  
For instance, cubic B-spline are  $C^2$  continuous.
- If one control point  $V_i$  is changed, it affects only that portion of the curve/surface (local modification).
- A relatively small number of points are required to define complicated geometries (Theodoracatos, 1993).
- B-spline provides the flexibility to design a large variety of shapes.
- B-spline representations can be easily combined in a B-rep based solid modeller (Piegl, 1995).
- B-spline representations have been an IGES standard since 1983.

The comparing results between parametric and B-spline representations for major analytic entities are listed in *Appendix 3*.



#### 4.4 The B-spline Data Structure

A part model can be represented by B-rep based solid modelling schemes where a body consists of faces, a face consists of loops, a loop consists of edges, and finally an edge consists of vertexes. As shown in Fig. 4.2, a solid model based on B-rep consists of topological and geometric elements. B-spline representations are concerned with the geometric elements. The more detailed explanation about B-rep based solid models is discussed in chapter 5. Herein, in this section, data structure related to B-spline representations is presented.



#### 4.4.1 The data structure of B-spline curve

The header of B-spline curve is stored and manipulated in dbCRV structure. In function calls involving putting and getting B-spline curves to and from the database of the proposed system, the standard C structure with the tag “CURVE” is used.

```
typedef struct dbCRV {
    int order;           // The order of a curve
    int no_knot;        // The number of knots
    float *knot;        // knot vectors
    int no_ctrlpt;     // The number of control points
    float *ctrlpt;     // Control points
} CURVE;
```

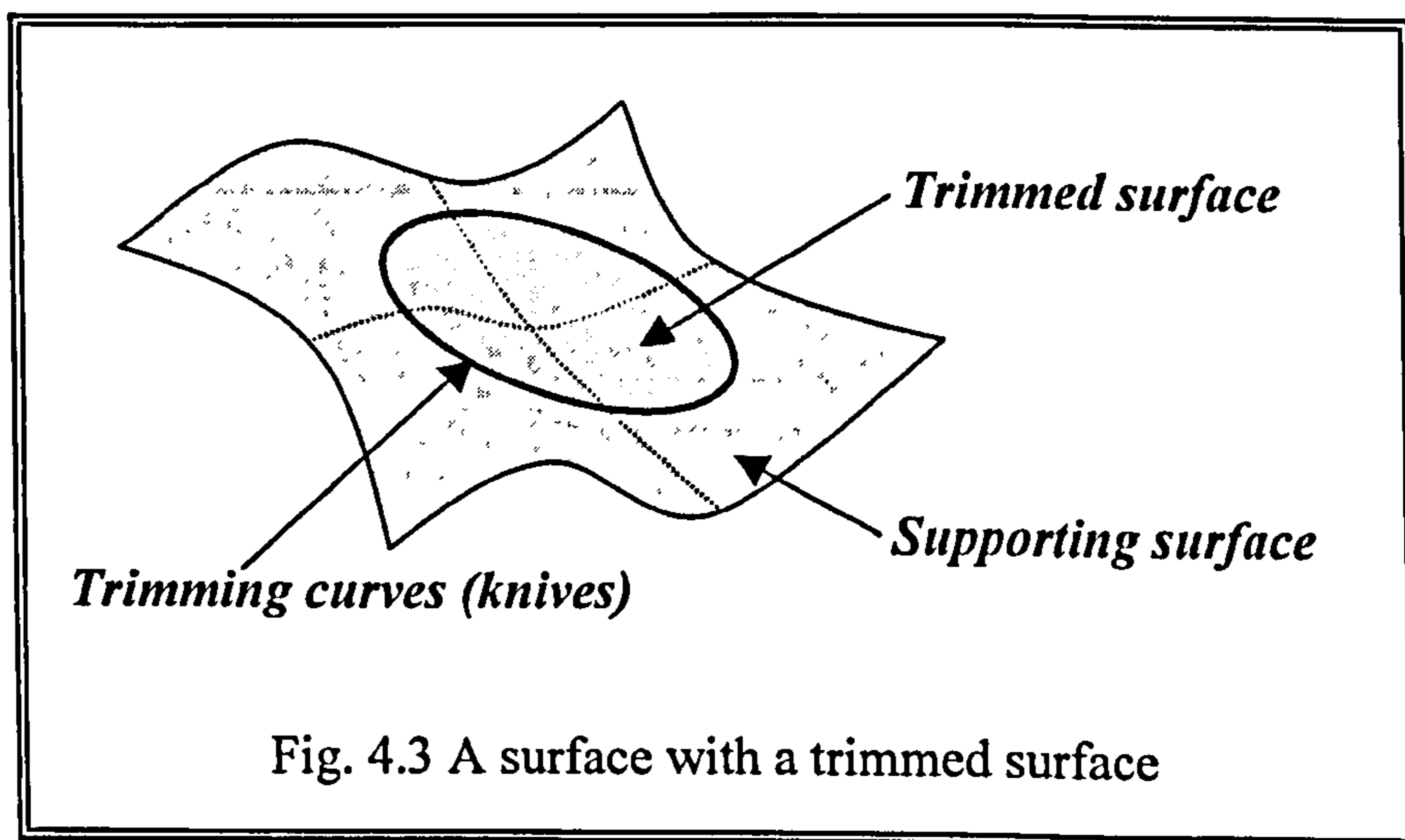
#### 4.4.2 The data structure of B-spline surface

The proposed dbSUR structure has ten elements: *orders*, *number of knots*, *knot vectors*, *number of control points* and *control points*, which have two directional value ( $u$  &  $v$ ) except the *weight*. The data structure of B-spline surfaces in the application is as follows.

```
typedef struct dbSUR {
    int u_order;        // u-dir order
    int v_order;        // v-dir order
    int no_uknot;       // The number of u-dir knot vectors
    float *u_knot;      // U-dir knot vectors
    int no_vknot;       // The number of v-dir knot vectors
    float *v_knot;      // V-dir knot vectors
    int no_uctrlpt;     // The number of u-dir control points
    int no_vctrlpt;     // The number of v-dir control points
    float *ctrlpt;      // Control point nets
} SURFACE;
```

## 4.5 Trimmed Surfaces

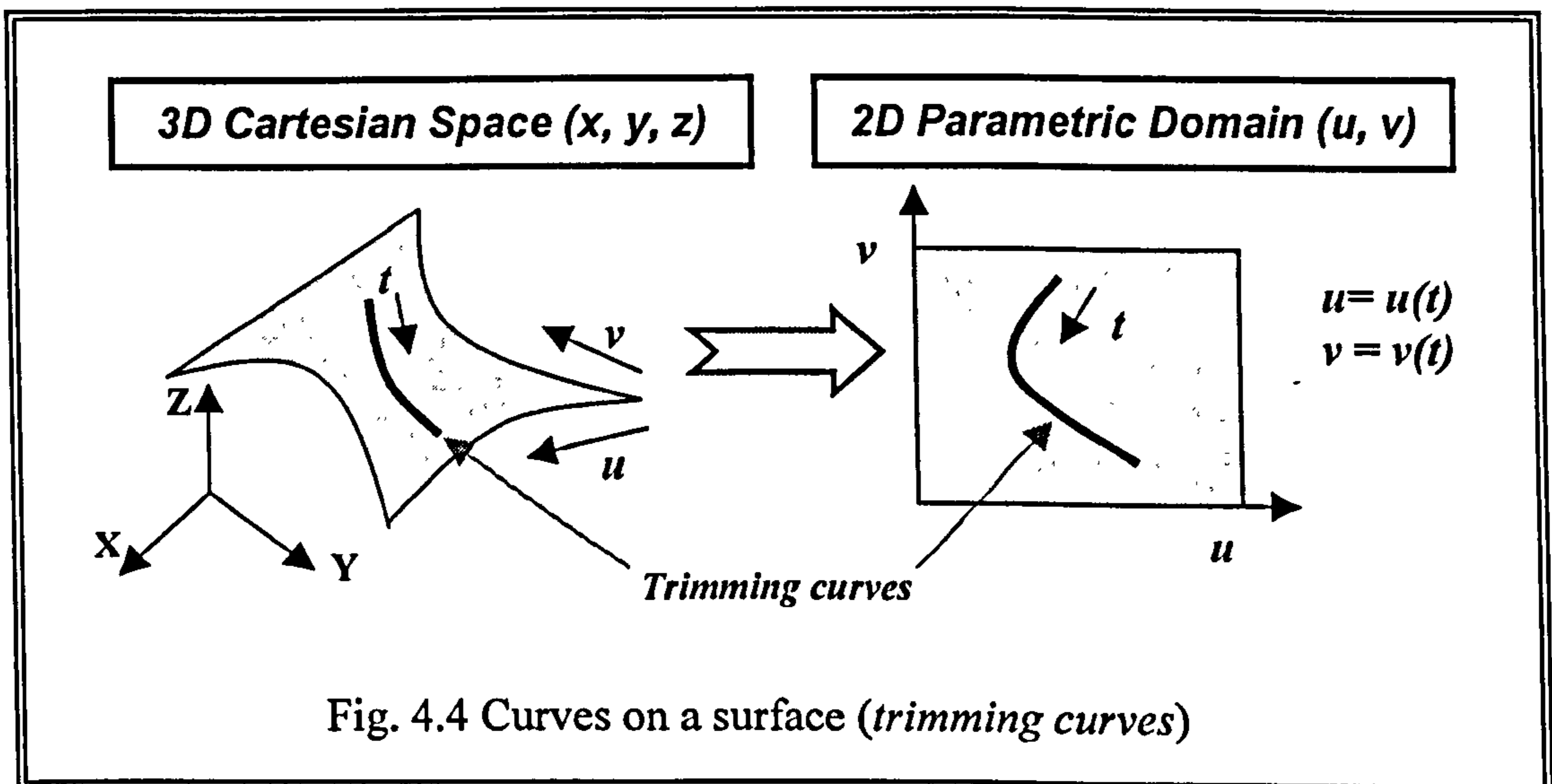
In general, the boundary curves of a generated surface will not coincide with the boundaries of the input point clouds. Thus, the surface regions outside interested portion need to be trimmed away by knife-curves defined in the parametric domain of the surface. A trimmed surface is a surface that is essentially a subset of a supporting surface, and it is defined by a set of trimming curves that describe the boundary of that part of the whole surface that it occupies (Jun, 1993). For instance, an equation for a surface along with the centre and radius of the circular region can be specified for a trimmed surface as shown in Fig. 4.3.



A surface provides a  $2D$  space ( $u$  and  $v$  domain) suitable for supporting the vector analytic representation of curves. The two parameters defining the surface also supply the coordinate grid on which embedded curves may be defined. It is helpful to view these curves as curves along which the surface was trimmed in order to leave the area of interest (Hu, 1997; Parasolid, 1998).



It is known that the treatment of solid objects with free form surfaces is one of the major problems. However, fitted free form surfaces in the ReTool system can be the internal and external boundary of extracted solid models in the B-rep based modelling. The techniques presented in this section will thus be useful to reconstruct B-rep solid models of parts containing faces with multiple loops and trimmed surfaces. These regions are represented by an analytical surface or a free form surface. As shown in Fig. 4.4, the trimming curves must be defined in the parameter space of the surface. Projection, iso-parametric and intersection curves are the examples of trimming curves which lie on the parameter space of a surface.



The trimmed surface and its trimming curves are formulated as follows. Consider a tensor product B-spline surface

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m V_{i,j} N_{i,p}(u) N_{j,q}(v) \quad (4.12)$$

and a given B-spline trimming curve  $C(t) = [u(t), v(t)]^T$  in  $u$ - $v$  parametric domain. Since  $u = u(t)$  and  $v = v(t)$ , the trimmed surface can be modified such that

$$S(u, v) = S(u(t), v(t)) = C(t) = \text{points on the curve, on the surface}$$

It clearly says that the trimmed surface equation can be represented by the equation of trimming curves. To determine the coordinates of points on this curve in the *3D Euclidean* space, first compute pairs of  $u$ ,  $v$  values for successive values of  $t$ . Substitute these  $u$ ,  $v$  values into the surface equation to obtain a sequence of  $x$ ,  $y$ , and  $z$  coordinates of points in object space as follows.

$$\begin{aligned} E(t) &= S(C(t)) \\ &= S(u(t), v(t)) \\ &= (x(u(t), v(t)), y(u(t), v(t)), z(u(t), v(t))) \end{aligned}$$

The direction of trimming curve determines which side of cut is retained. In this research, the area on the left of the cut is the area maintained as shown in Fig. 4.5.

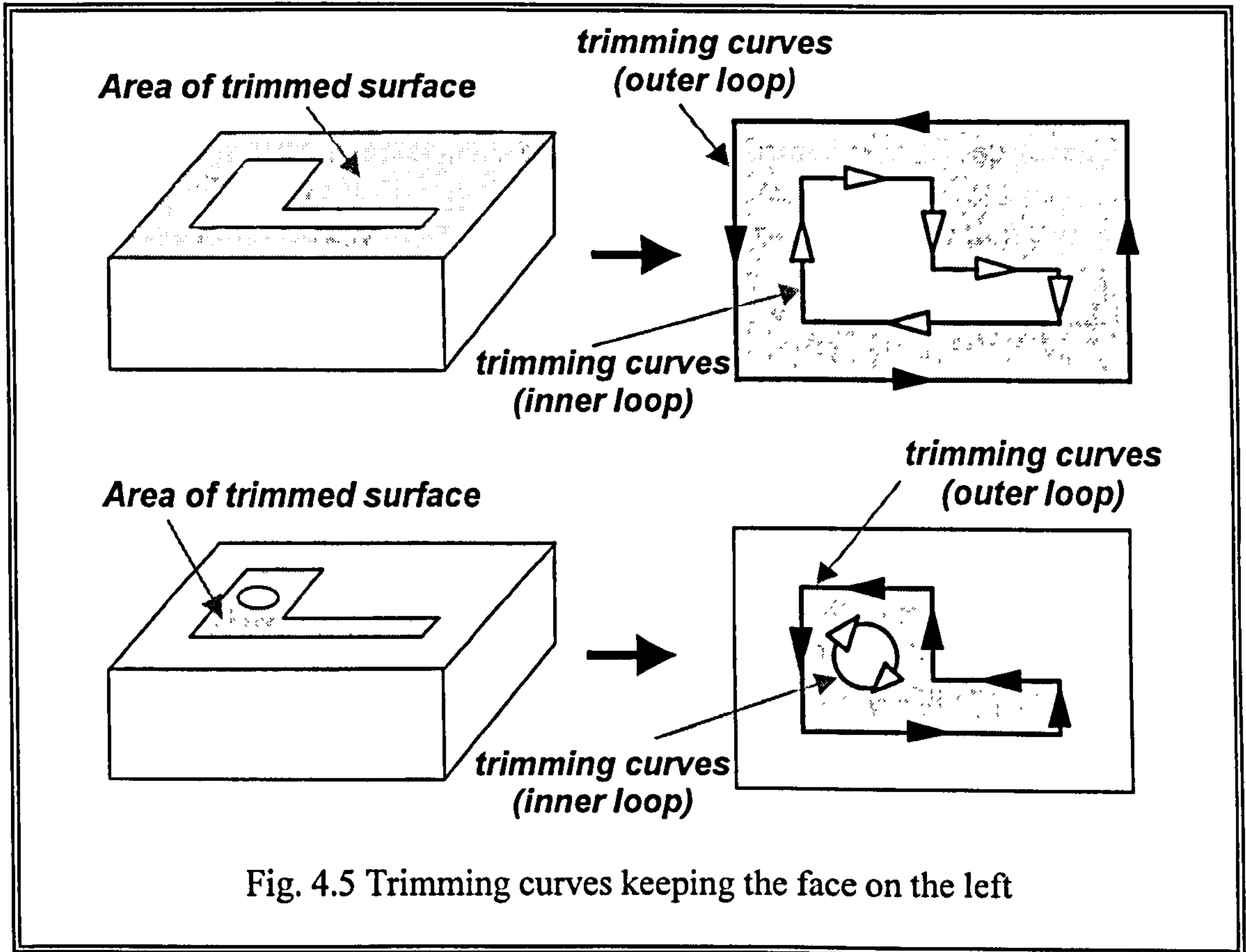


Fig. 4.5 Trimming curves keeping the face on the left



#### 4.6 B-spline Entities within Standard File Formats

The main goal of this chapter is to digitise a free-formed part, fit surfaces and import them into other CAD/CAM systems for further modifications or next downstream activities. IGES and STEP are the most widely adopted neutral file format for exchanging product data among today's CAD/CAM systems (Piegl, 1995). B-spline representations can be specified both in the IGES and STEP. In the ReTool system, the reconstructed surfaces are exchangeable with other systems through IGES file formats for the entity type 128 (B-spline surface).

In this section, the definitions of B-spline curves within these two standards are compared as shown in Table 4.1. More detailed explanation regarding IGES file format is given in chapter 8.

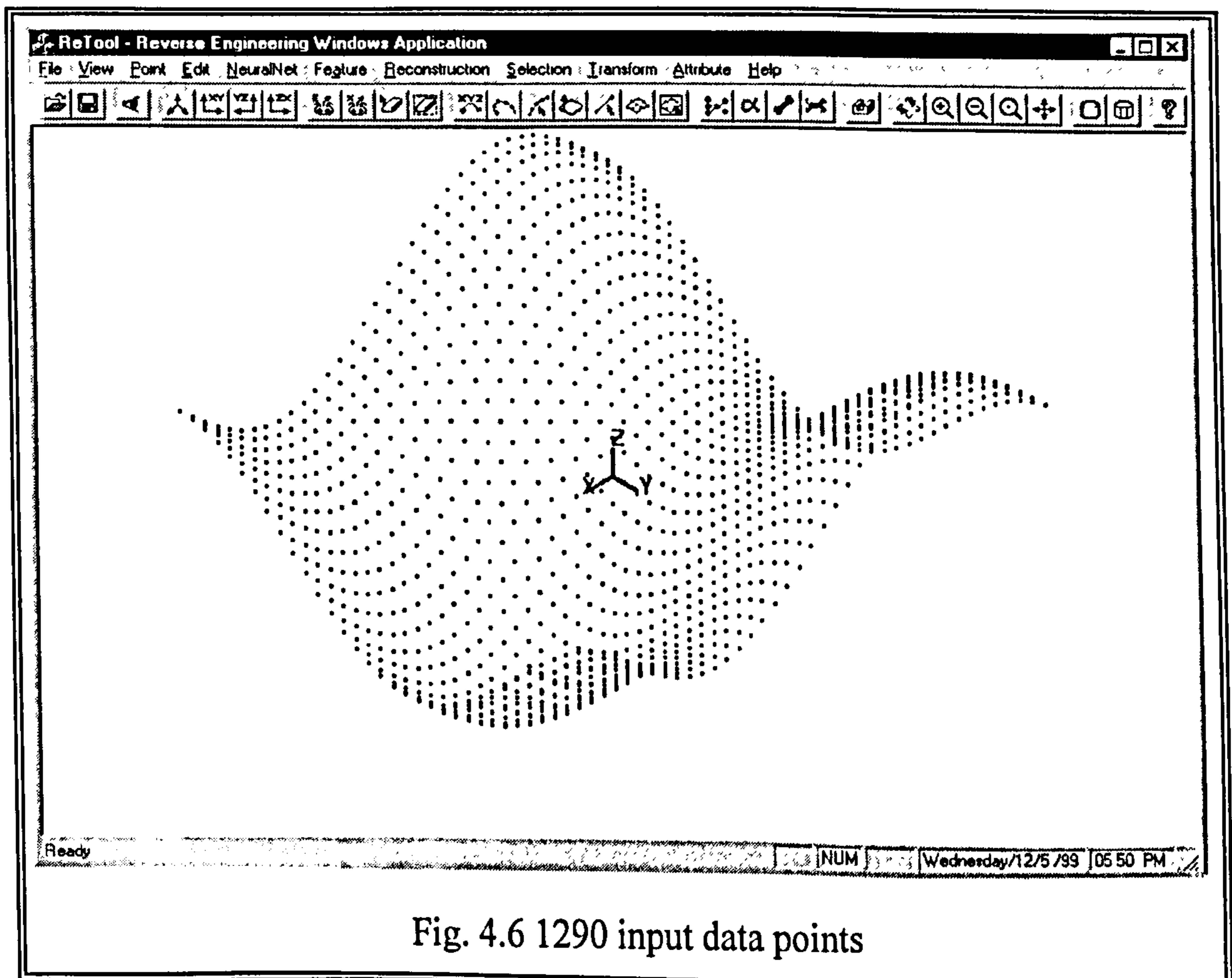
Table 4.1 Comparisons of B-spline curve definition in IGES and STEP

	<i>IGES</i>	<i>STEP</i>
degree	$p$	$p$
no. of control points	$n+1$	$n+1$
control points	$V_i$	$V_i$
weights	$w_i$	$w_i$ (must be <i>positive</i> )
knot vector	$U(u_i \leq u_i \& m+1 = n+p+2 \text{ knots})$	$U(u_i \leq u_i \& m+1 = n+p+2 \text{ knots})$
start/end parameter	$s_0$ and $s_1$	
other information	<i>Planar/nonplanar, rational/nonrational, periodic/nonperiodic</i>	<i>Open/closed, whether or not self-intersecting</i>
tag	<i>Line/circular/conic arc</i>	<i>Polyline/circle/conic</i>

## 4.7 B-spline Surface Fitting Examples

The developed B-spline fitting module is of importance because it is widely applied in the industry for design of free form surfaces. In order to verify the effectiveness of the implemented B-spline module, a simulation for fitting a complex surface is tested. The data points used in the example were generated by a *bivariate sinusoidal* analytic equation,  $z = \sin(x) + \cos(y)$ . This kind of shape is less commonly encountered in consumer products but provide a slightly stricter test of the module.

In the example, the data points, as shown in Fig. 4.6, consists of 1296 points, i.e.: a total of 36 sections (scan lines) parallel to the  $x$ - $z$  plane with each having 36 points were created.





The range of the amplitude was from  $-20$  to  $20$  and the  $x$ - and  $y$ -unit were both  $0-60$ . The degree of B-spline basis function was set at three for all surface patches, since this is commonly used by most commercial CAD/CAM systems. The only user intervention to create a surface based on a point cloud was to define four corners using a mouse 'window' function (*segmentation*).

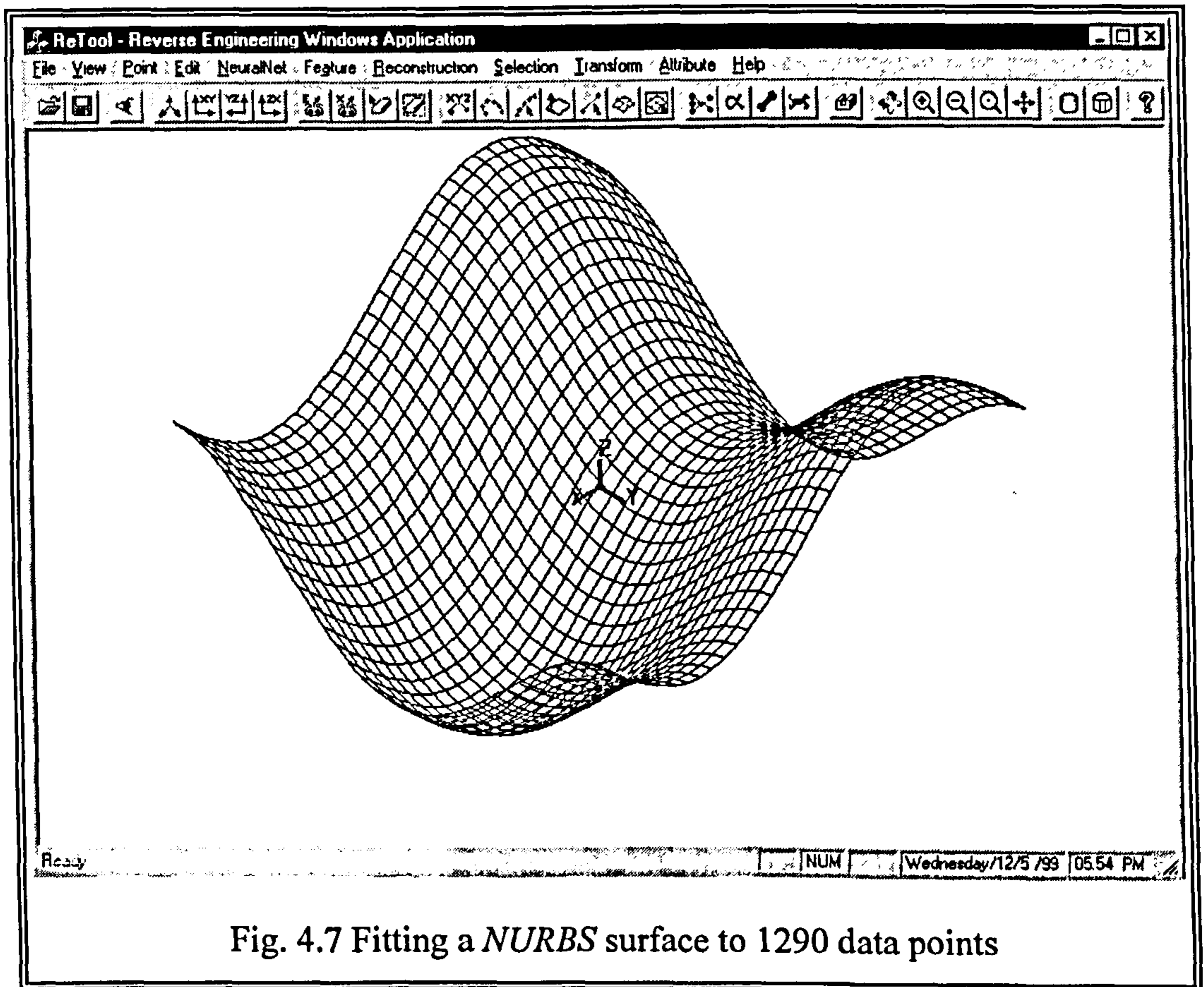


Figure 4.7 illustrates the reconstructed surface and its shaded image is shown in Fig. 4.8. Using the B-rep modelling techniques, an enclosed B-rep solid model can be created with reconstructed free form surfaces. Figure. 4.9 shows some reconstructed surfaces with defining sub-areas of the part.



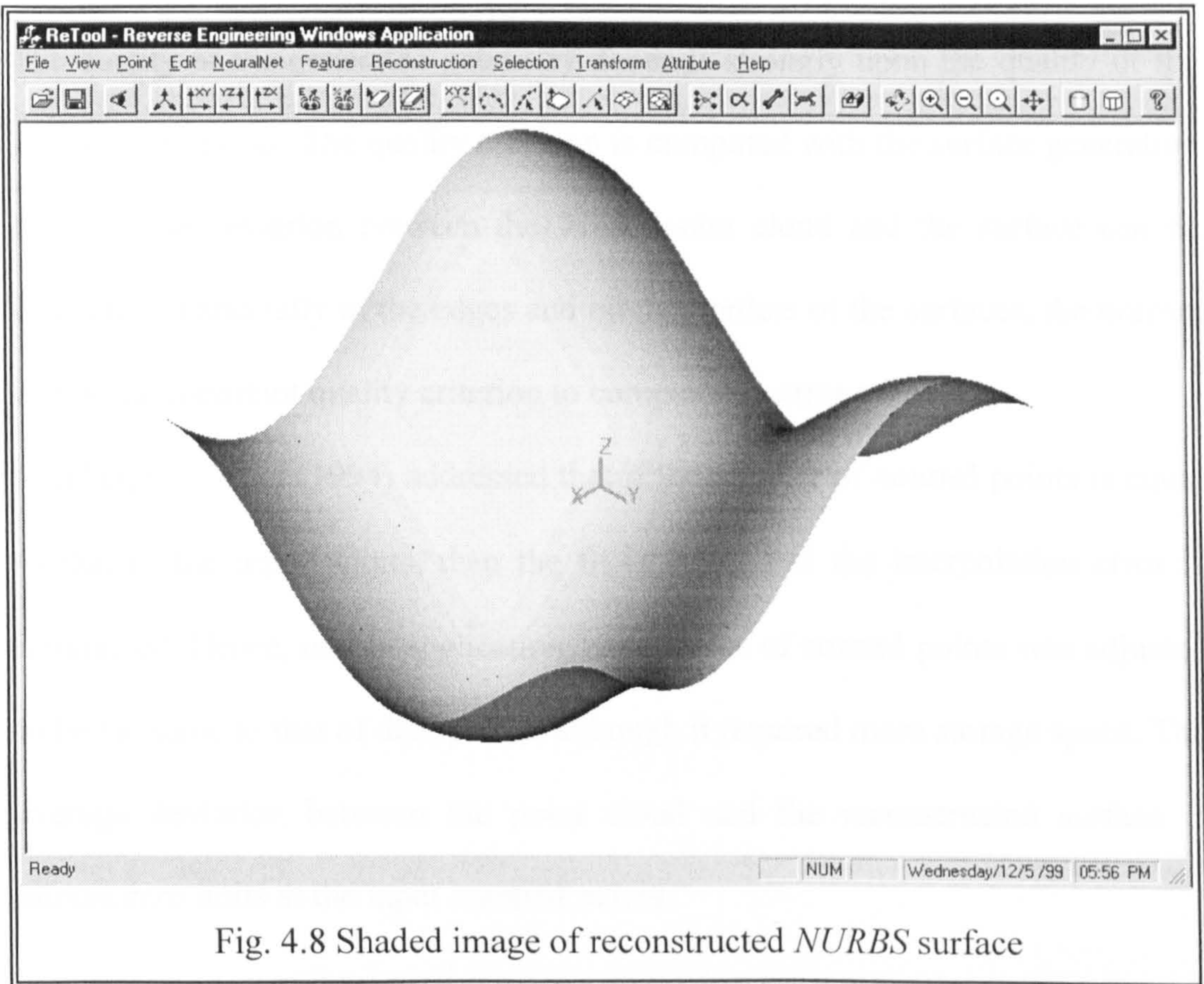


Fig. 4.8 Shaded image of reconstructed *NURBS* surface

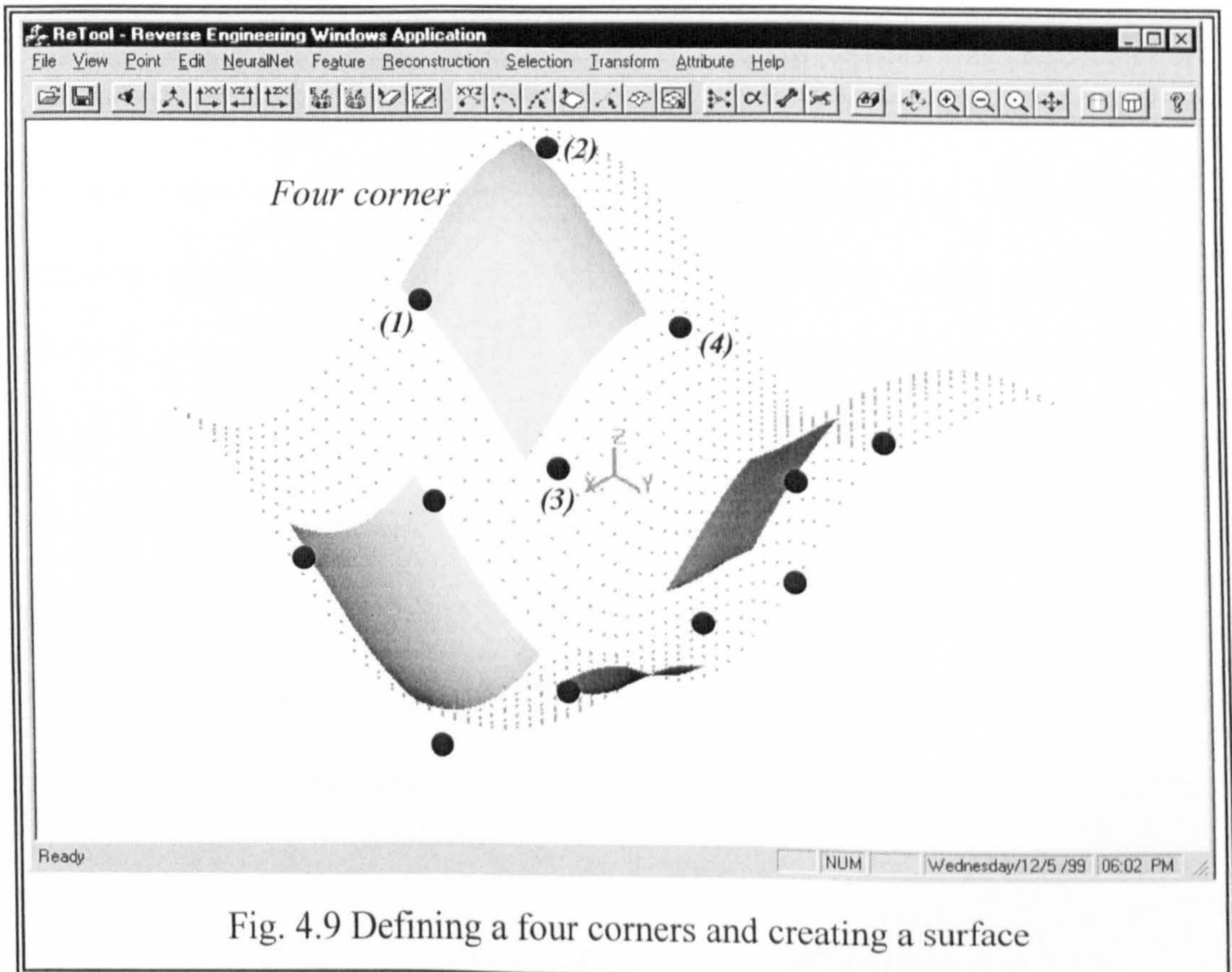


Fig. 4.9 Defining a four corners and creating a surface



The quality of the generated geometry depends strongly upon the quality of the scanned point data. The quality criterion is computed with the surface generation error. These deviation between the whole point cloud and the surface can be calculated. Especially in the edges and on the borders of the surfaces, the normal can be an important quality criterion to compute the error value.

Mukherjee S. *et al* (1994) addressed that if the number of control points is equal to that of the input points, then the fit is better and the interpolation error is minimised. Hence, in this application, the number of control points was adjusted to be the same to that of data points, although it required more storage space. The average deviation between the point cloud and the reconstructed surface is almost *zero* units at the input scanned points.

## Chapter 5

# Boundary Representations for Reconstructed Parts

How is a physical object represented in computers? Objects can be presented by using one of the following three models (Toriya, 1993): wireframe model, surface model, or solid model.

*Wireframe model* represents a 3D object only with edges and vertices. These structures can easily be represented in a computer, because only a small amount of memory capacity is necessary and data can be quickly accessed. However, an object represented by the wireframe model cannot be interpreted uniquely because of visual ambiguity. The wireframe model is useless for obtaining volume, surface area, or centre of gravity data.

The *surface model* has surface data in addition to wireframe data, so that computer can calculate accurately the  $x$ ,  $y$  and  $z$  coordinates of any point on the surface. Most of the existing 3D CAM systems are based on the surface model.

The surface model of an object is merely a group of surfaces. Hence, it is not



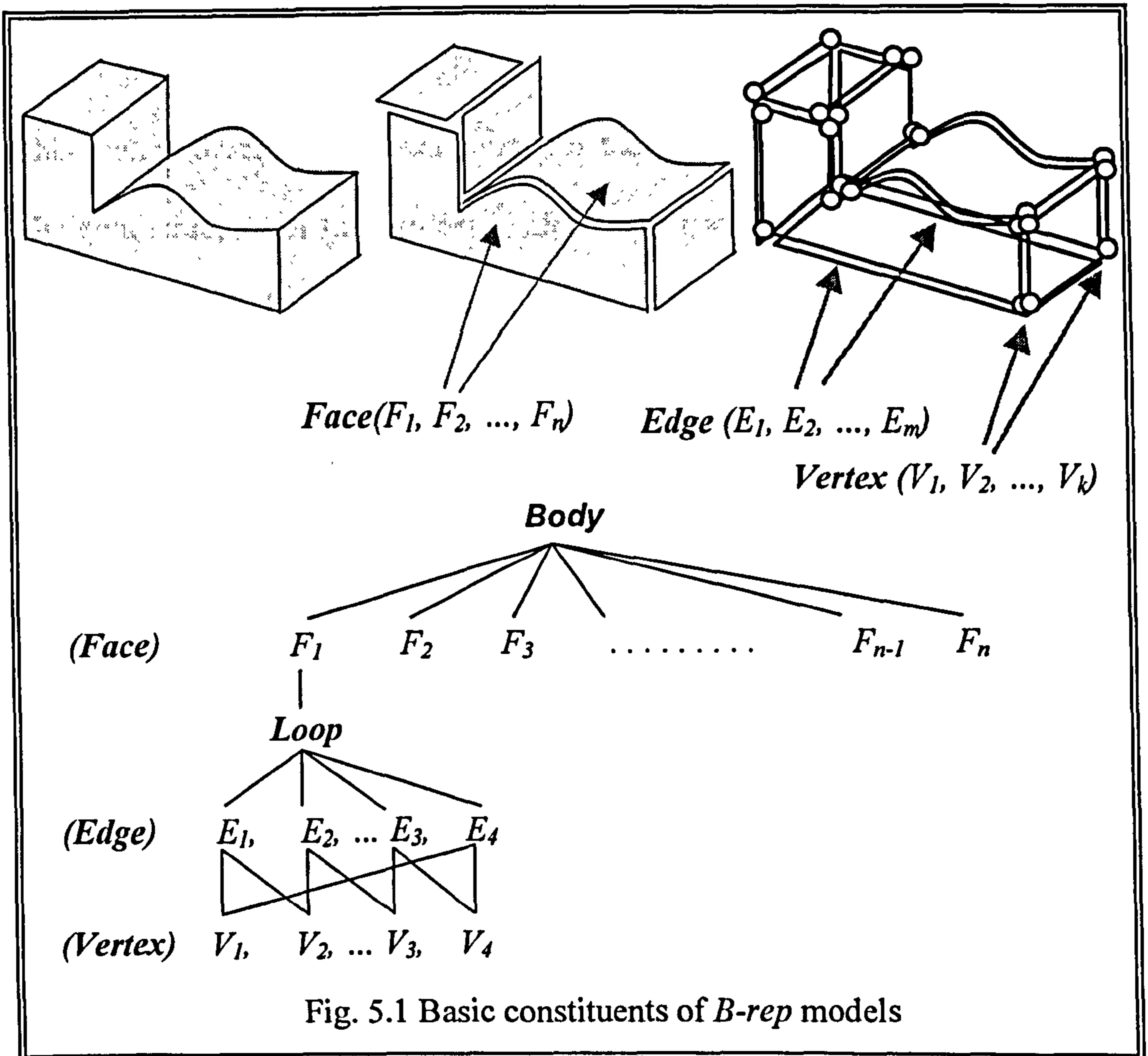
well suited for representing solid objects because it does not contain data for its contents. Consequently, mass properties calculation is not possible. In some situations, however, it has the advantage of being less demanding computationally and may achieve greater accuracy.

*Solid modelling* is one of the most advanced modelling techniques used in geometric modelling systems (Gu, 1995a). Like surface models, solid models define the edges and the surfaces of parts. The difference is that a solid model contains information about the interior of the model. It can provide mathematically unambiguous and complete information for physical objects (Hoffmann, 1989). Solid models can be used for interference check or to compute the amount of material removed during machining process. Also, it can calculate properties such as volume, weight, centroid, and moment of inertia. Although there are several representation schemes evolved and used in a solid modelling system, the most popular representation schemes are boundary representations (B-rep) and constructive solid geometry (CSG) techniques (Mantyla, 1985). A B-rep solid model is usually represented in a graph format in which low level entities of the model such as vertices, edges, and faces are represented explicitly. However, CSG models hold no explicit information concerning the geometry of the part, but instead describes how to obtain the part from a series of addition and subtraction operation performed on geometrical entities. Many commercial systems currently use a hybrid approach, converting the CSG working model to a B-rep definition, and stored both models in memory.

In this research, extracted features are represented as a volume contained in a set of surfaces together with topological information that defined the relationships between the surfaces. The proposed data structure of the feature is closely similar to the B-rep based representation. Therefore the aim of this chapter is to outline B-rep based solid modelling techniques.

## **5.1 The Overview of B-rep based Modelling**

A B-rep model represents a solid by a representation of its bounding surfaces. These surfaces are closed and have orientations indicating that it is possible to distinguish between the two sides of the surface. Thus, a B-rep solid model will have an interior and an exterior region. A face is a bounded region of a single planar, quadratic, free form or other surface. The bounded region of the surface that forms the face is defined by several bounding curves that lie on the face. The bounding curves of each face are represented by edges and each edge is bounded by two vertices as shown in Fig. 5.1.



Thus, topologically, the B-rep model of a part is composed of faces, edges and vertices of the part linked together with maintaining the topological consistency of the model. A B-rep model has to satisfy the following conditions (Hoffmann, 1989; Jablokow, 1993): The set of faces form a complete skin of the solid with no missing part, and faces do not intersect each other except at common vertices or edges. The boundaries of faces (edges) do not intersect themselves. These conditions disallow self-intersecting and open solids.

B-rep models can represent a wide range of objects relatively fast but the data structure is complex, and it requires a large memory space.



## 5.2 Data structure of B-rep

This section describes the data structures that store B-reps based solid models. The data structure of the geometric elements of a model can be automatically determined from the mathematical representations of the elements. This section therefore focuses on the data structure of the topological elements, which specify how boundary surfaces of a model are connected.

### 5.2.1 Euler–Poincare' law

In a B-rep model, the geometry is created and the topological validity of the model simultaneously ascertained. It is customary to bundle all information of the geometry of the entities under the term geometry of a boundary model, and their adjacencies and interconnections under the term topology (Mantyla, 1985).

As shown in Fig. 4.2, the topology contains the information on the relation of the entities, i.e. how the faces, edges and vertices are connected together. The geometric information contains the face and edge equations, and vertex coordinates. Topology is created by performing *Euler–Poincare'* (*E-P*) law and geometry is created by performing Euclidean calculations, respectively.

To make sense the geometric and topological validity of boundary models, the *E-P* law applies to the manipulation of the topological relationships:

$$V - E + F = 2 \quad (5.1)$$

Where  $V$  is the number of vertices,  $E$  is the number of edges and  $F$  is the number of faces. The *E-P* law is used for creating and consistently modifying

the topology of manifold object surfaces. They can create closed surfaces, and modify these surfaces by adding or deleting faces, edges, and vertices. The law is not restricted to solid with planar faces, so that the same law can be applied to a curved object. While B-rep systems store only the bounding surfaces of the solid, it is possible to calculate volumetric properties such as mass properties by virtue of volume integrals to surface ones (Zeid, 1991).

### 5.2.2 Manifold and non-manifold

A physical real world object is manifold. That is, geometrically, if the topology of a solid model satisfies the following two conditions, the solid is called *manifolds*:

- A solid has no part which shares only edges.
- A solid has no part which shares only vertices.

If a part does not satisfy above conditions, the solid is called *non-manifold*.

Figure 5.2 gives an example of non-manifold objects where edge  $lm$  or vertex  $v$  is the only part to share.

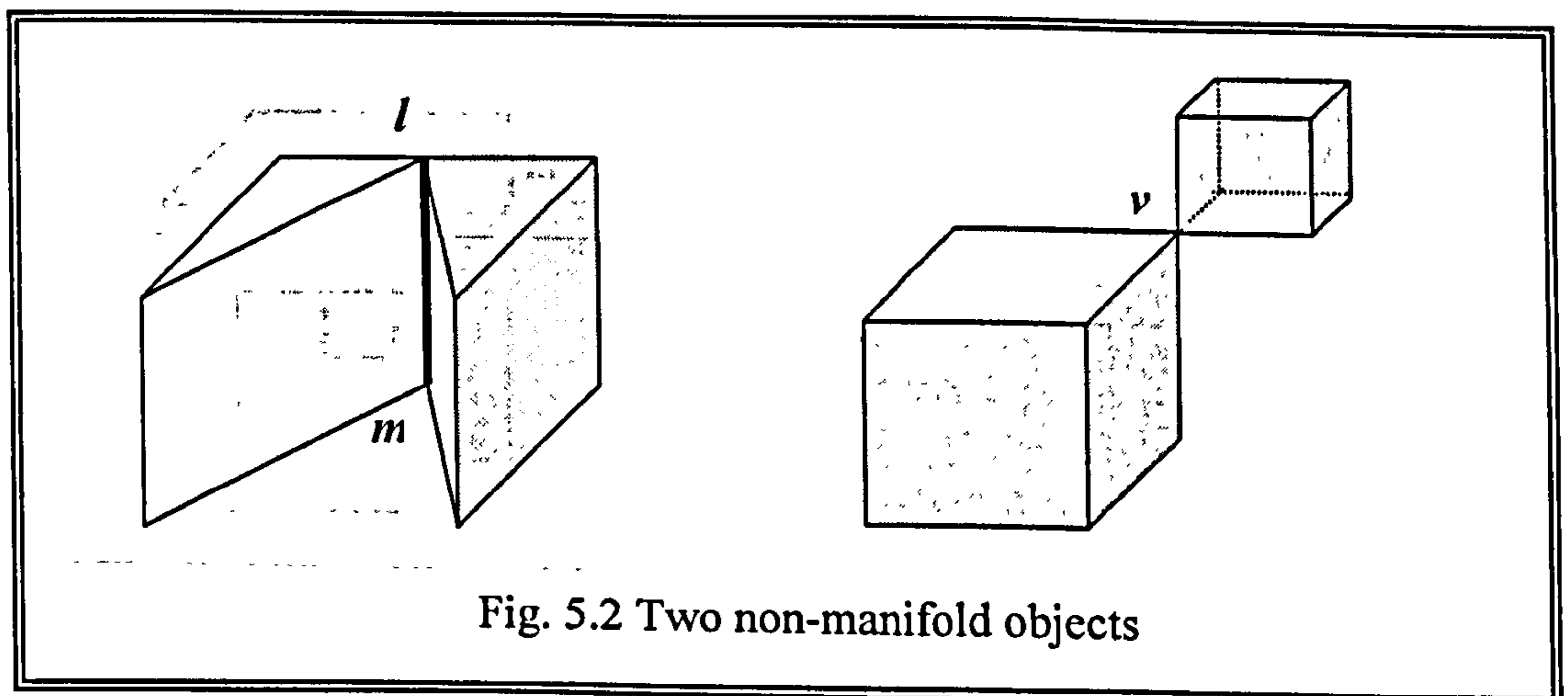


Fig. 5.2 Two non-manifold objects

### 5.2.3 B-rep model representations

There have been three kinds of topological data structures in B-rep that are currently used to define a geometric model: *winged-edge structure*, *half-edge structure* and *radial-edge structures*.

The winged-edge structure (Baumgart, 1972; Mantyla, 1985) shown in Fig. 5.3 is commonly used data structure for B-rep based solid, which can be uniquely defined for each edge of a solid.

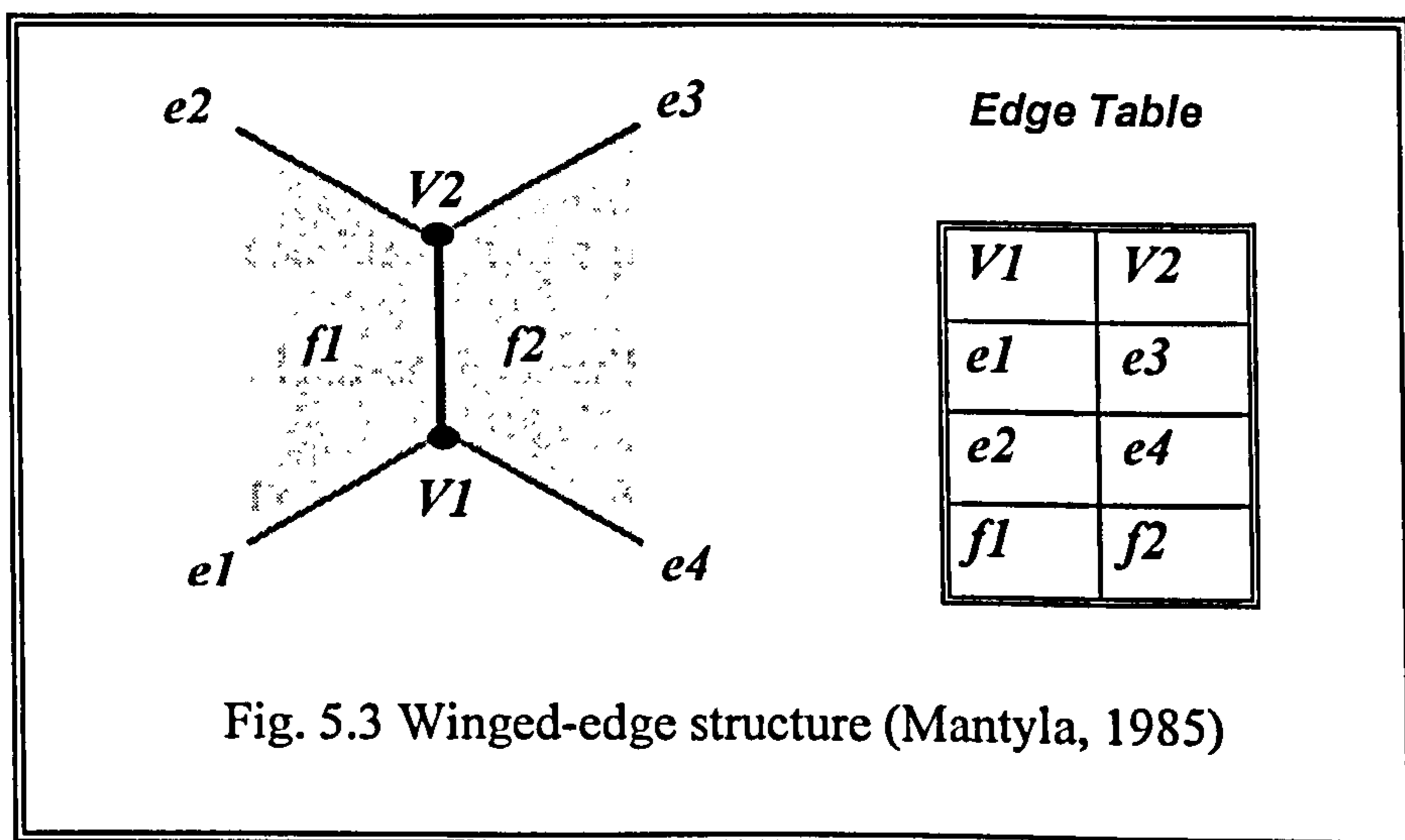
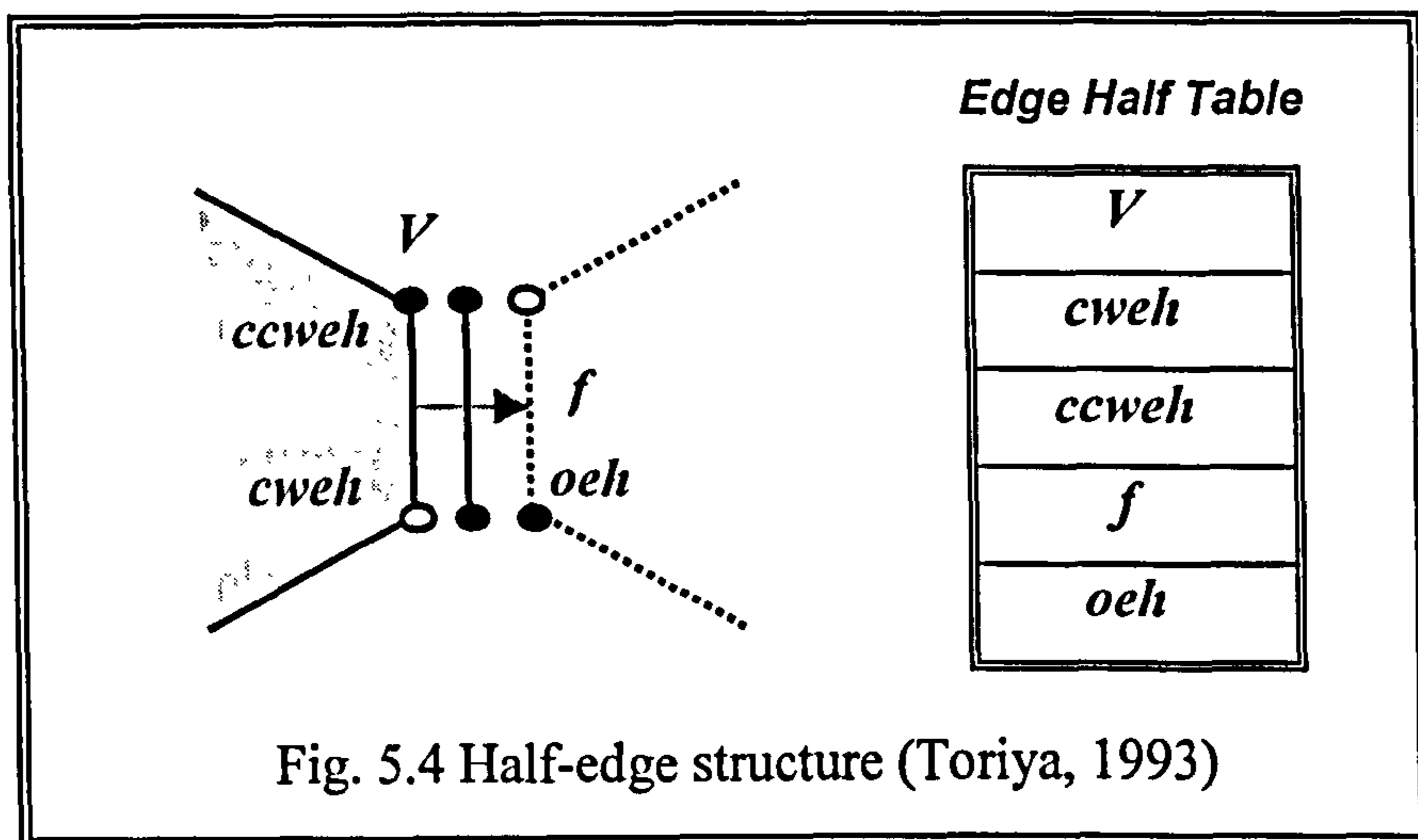


Fig. 5.3 Winged-edge structure (Mantyla, 1985)

As this figure shows, one edge always connects to two vertices ( $v1$ ,  $v2$ ), two faces ( $f1$ ,  $f2$ ), and four winged edges ( $e1$ ,  $e2$ ,  $e3$ ,  $e4$ ). This structure is specifically for the compact storage of part geometry. Although fixed-length data can be stored for each edge, winged-edge structure cannot be applied to a state where there is no edge. This is because it is an edge-based boundary model. The B-rep models of reverse engineered parts implemented in this application are similar to the winged-edge structure using loops.

The half-edge structure is based upon the separation of an element from either side of an edge used in the winged-edge structure as shown in Fig. 5.4.





The 'right' half is considered to be positively oriented, whilst the 'left' half is negatively oriented. Half-edge describes one line segment of a loop. It consists of a pointer to its parent loop and a pointer to the starting vertex of the line segment in the direction of loop. The final vertex of the line segment is available as the starting vertex of the next half-edge. This structure can represent the state where only one vertex ( $V$ ) exists in space. Both winged-edge and half-edge structure can only be employed to represent manifold parts (Woo, 1985).

Weiler K. (1988) proposed the radial-edge structure to represent both manifold and non-manifold parts. A new topological element *use* was introduced to allow non-manifold objects. Although this structure requires more storage than winged-edge structure and half-edge structure, the associativity information is of importance for use by various functions of an integrated modelling system.

However, since *RE* parts are always concerned with manifold parts, non-manifold modelling schemes were not considered in this application.

### 5.2.4 Topological data structure

The data structures of the four topological elements used are *Body*, *Loop*, *Edge* and *Vertex* structure. A *Body* means a solid, and its data structure was described using the *struct* of C-language. A *Body* data structure handles the number of loops, edges and vertices representing the *Body*. It also stores pointers to the loop table, edge table and vertex table which stores associative loops, edges, and vertices respectively as follows.

```
typedef struct dbBody {  
    int loop_number;  
    int edge_number;  
    int vertex_number;  
    struct dbLoop *loop_table;  
    struct dbEdge *edge_table;  
    struct dbVertex *vertex_table;  
};
```

In the *Loop* data structure, one of the attached edges to the loop is stored. There are two kinds of loop in B-rep models: outer and inner loop. The relationship between an outer loop and an inner loop has been shown in Fig. 4.5. An outer loop is a counter-clockwise connecting curve along to the outer boundary of a face whilst an inner loop is clockwise curve along to an inner boundary. The instance of *Loop* structure is given as below.

```
typedef struct dbLoop {
    struct dbLoop *outer_loop;
    struct dbLoop *inner_loop;
    struct dbEdge *edge; //one of the edges connecting a loop
    struct dbSUR *sur; // pointer of geometric data
    struct dbLoop *next;
};
```

In the structure of *dbEdge*, two vertices, two connected loops and four winged-edges are stored as shown in Fig. 5.3. Instance *\*crv* points the address of curve which represents the geometric data of the edge as follows.

```
typedef struct dbEdge {
    struct dbLoop *left_loop;
    struct dbLoop *right_loop;
    struct dbVertex *start_vertex;
    struct dbVertex *end_vertex;
    struct dbEdge *start_left_edge;
    struct dbEdge *end_left_edge;
    struct dbEdge *start_right_edge;
    struct dbEdge *end_right_edge;
    struct dbCRV *crv; // pointer of geometric data
    struct dbEdge *next;
};
```



Finally, the structure stores any one of the edges, *\*edge*, attached to the vertex, and the coordinates of a vertex can be accessed through the *\*pnt* that points the geometric data. The structure is as follows.

---

```
typedef struct dbVertex {  
    struct dbEdge  *edge; //one of the edges connected to a vertex  
    struct dbPNT   *pnt;  // pointer of geometric data  
    struct dbVertex *next;  
};
```

## Chapter 6

# Feature-based Technology in Reverse Engineering

Part of the difficulty in geometric interpretation is that the methods of wireframe, surface, and solid modelling are semantically not very rich. There is no information, for example, to say that a collection of lines and arcs, or of cylinders and cones, represents a drilled and tapped hole in the machining application. A trained human observer can, nevertheless, interpret such features on a drawing without difficulty.

Feature-based technology is a more intelligent way to design CAD models. It is a means to abstract the level of design by working with high-level shapes instead of low-level primitives; giving a better understanding of the part geometry as well as its functionality. That is, engineering meaning is included in feature based models.

A part may contain many instances of the same kind of feature, especially on prismatic parts. It is desirable to describe a part in terms of the shape features to perform particular functions since features can capture higher level engineering structures of the part. Only feature-based model can directly support many engineering applications which functional information is crucial to shape data such as holes, slots, pockets, steps, and so forth (Gu, 1995a). Under the feature-based modelling scheme, it is relatively straightforward to make modifications or generate effective process plans automatically (Thompson, 1995). In addition, for true integration of CAD with CAM, feature-based technology is the efficient way to understand design part geometry (Devireddy, 1999).

Hence, extracting the relevant feature information is crucial for the subsequent stages of an automated RE process. Extracting features directly from scanned point set can be the first step to make a fully automated RE application.

The aim of this chapter is to discuss the state-of-the-art feature-based techniques<sup>1</sup>. In particular, all previous feature recognition processes and the basic prismatic features that commonly occur in engineering parts are discussed in depth.

---

<sup>1</sup> If *feature-based technology* is technically developed enough, the other modelling methods such as wireframe, surface, and solid modelling techniques are unnecessary in many engineering fields including *RE*. Unfortunately, in the context of *feature-based technology*, we have a long way to go for making a fully automatic engineering system with the *feature-based technology* in *CAD/CAM*. That is why both the *B-rep* based technology and the *feature-based technology* were employed in this research to represent physical objects.



## 6.1 Feature-based Technology

Feature-based technology is an extension of the solid modelling schemes. It has evolved in response to industry needs in design and manufacturing, particularly the need to reduce product development time. It is regarded as a key enabling technology for the implementation of concurrent engineering and CAD/CAM integration in the future (Shah, 1995). Currently, several CAD systems support features to improve the design process and the data exchange to manufacturing applications such as process planning and NC programming (Tonshoff, 1996). No general-purpose commercially available feature-based modeller has come up in the market. Some progress has been made only in research modellers instead.

A review of the literature reveals that there have been a few investigations about feature-based reverse engineering. Thompson, W. *et al* (1996) proposed a prototype of a RE system called REFAB (Reverse Engineering – FeAture-Based) which uses manufacturing features as geometric primitives. In order to extract features, the system is completely dependent on user intervention.

Motavalli, S. *et al* (1996) proposed an object-oriented database structure for the RE system and feature-based design using several orthographic views.

## 6.2 Definition of Feature

Many definitions have been proposed in the literature for a feature, but most of them are either too general or too application-oriented (Prebhakar, 1992). Feature-based models must be limited within the well-bounded domain defined by available generic features. Features can be classified as functional features, design features, or manufacturing features. All of these classifications interpret the same feature in different ways. In the geometric modelling view, a feature represents the engineering meaning of the geometry of a part or an assembly. The engineering meaning may involve the function of the feature, or how it can be created, or how the feature behaves in various situations. Features are grouping of geometric or topological entities that need to be referenced together.

Henderson, M. (1990) described features as *'geometric and topological patterns of interest in a part model and which represent high level entities useful in part analysis'*.

Shah, J. (1992) defined features as *'representing the engineering significance of the geometry of a part associate certain useful attributes and knowledge'*.

A feature is defined by Devireddy, C. *et al* (1999) as *'a geometric form or entity whose presence or dimensions are required to perform at least one CIM function and whose availability as a primitive permits the design process to occur'*.

The shape of a feature may be expressed in terms of dimension parameters and enumeration of geometric and topological entities and relations. A major function of features is to create the associativity of entities in a product definition (Shah, 1995). A crucial side benefit of feature representation is that algorithms are available for automatically computing the mass and section properties of feature models.

Other important characteristic of features is that all feature property can be separated into generic properties and specific properties. For instance, all pockets have certain generic properties regardless of their size, specific location and orientation like the depression characteristic. On the other hand, every pocket instance in a part also has specific properties such as position, size and orientation.



### 6.3 Advantages of Feature-based Approach

A major advantage of the feature-based approach is that it can provide an additional layer of information in CAD applications to make them more useful. This approach provides the designer with a set of features that have significance for either design or manufacturing, and contains a well-defined engineering meaning such as geometric characteristics of a part. They may be thought of as packets of related characteristics of part (Shah, 1995).

Other advantages of features are as follow (Shah, 1992; Thompson, 1996),

- Many sophisticated mechanical parts can be compactly represented by features.
- The use of features as primitives can substantially increase the accuracy of generated models.
- It is straightforward to import the reconstructed feature models into existing commercial feature-based CAD systems. Some examples of the feature-based modelling systems are: *DICAD* (Grabouski, 1990), *GEKO* (Bauert, 1990), *IIICAD* (Pratt, 1990), *ASU Feature Testbed* (Shah, 1990; Wang, 1991), and *GEONODE* (Bronsoort, 1993).
- Designers can design directly in terms of features, such as blind holes and rectangular pockets.
- Due to associativity between defined entities, design change propagates through the entire model.

## **6.4 Feature Relationships**

The following characteristics among features are crucial to develop a feature-based application (Shah, 1995).

### **6.4.1 Parameter inheritance**

When a new feature is defined on a model, some of the parameters of the new feature may already be determined. For instance, a through hole in a block must have a length equal to the depth of the block. Thus there needs to be a mechanism for features to derive certain parameters from the definition of other features. This is referred to as parameter inheritance.

### **6.4.2 Adjacency relationships**

When a feature is deleted or modified, it may cause problems for its neighbour. Hence there needs to be information regarding surrounding of a feature.

### **6.4.3 Placing of features**

When creating certain kinds of features, the entity on which the feature is to be placed needs to be identified. Typical positioning methods may fall under the following three categories:

- Position a feature in world coordinates (this method is used in this work).
- Position a feature adjacent to (tangent to) face/edge/vertex.
- Position feature faces/edges/axes on or along faces/edges of parent feature.

## 6.5 Feature Database in the Application

The C-language structure of the generic feature is defined in two levels. A parent structure contains field pointers that are commonly needed across all generic features and a child structure of a generic feature contains parameters specifying each feature as follows.

```
// Parent structure for features
```

```
typedef struct Pfeature {
    char *feature_name;    // Points to name of atomic feature
    PFeature *parent_feature; // Points to parent feature
    CFeature *child_feature; // Points to child feature
    PFeature *next_feature; // Points to next brother feature
    PFeature *prev_feature; // Points to previous brother feature
    float *origin;        // Points to position vector of origin
    float *orientation;   // Points to orientation vector
    float Depth (or Height); // feature height or depth
}
```

```
// An example of child structure for a hole feature
```

```
typedef struct Tapered_Through_Hole {
    float TopDiameter,    // Top diameter of hole
    float BottomDiameter, // Bottom diameter of hole
}
```

Feature hierarchy denotes the topological data between associative features as shown in Fig.6.1. The nodes denote instances of the generic features defined previously and the link arrows denote their relationships.



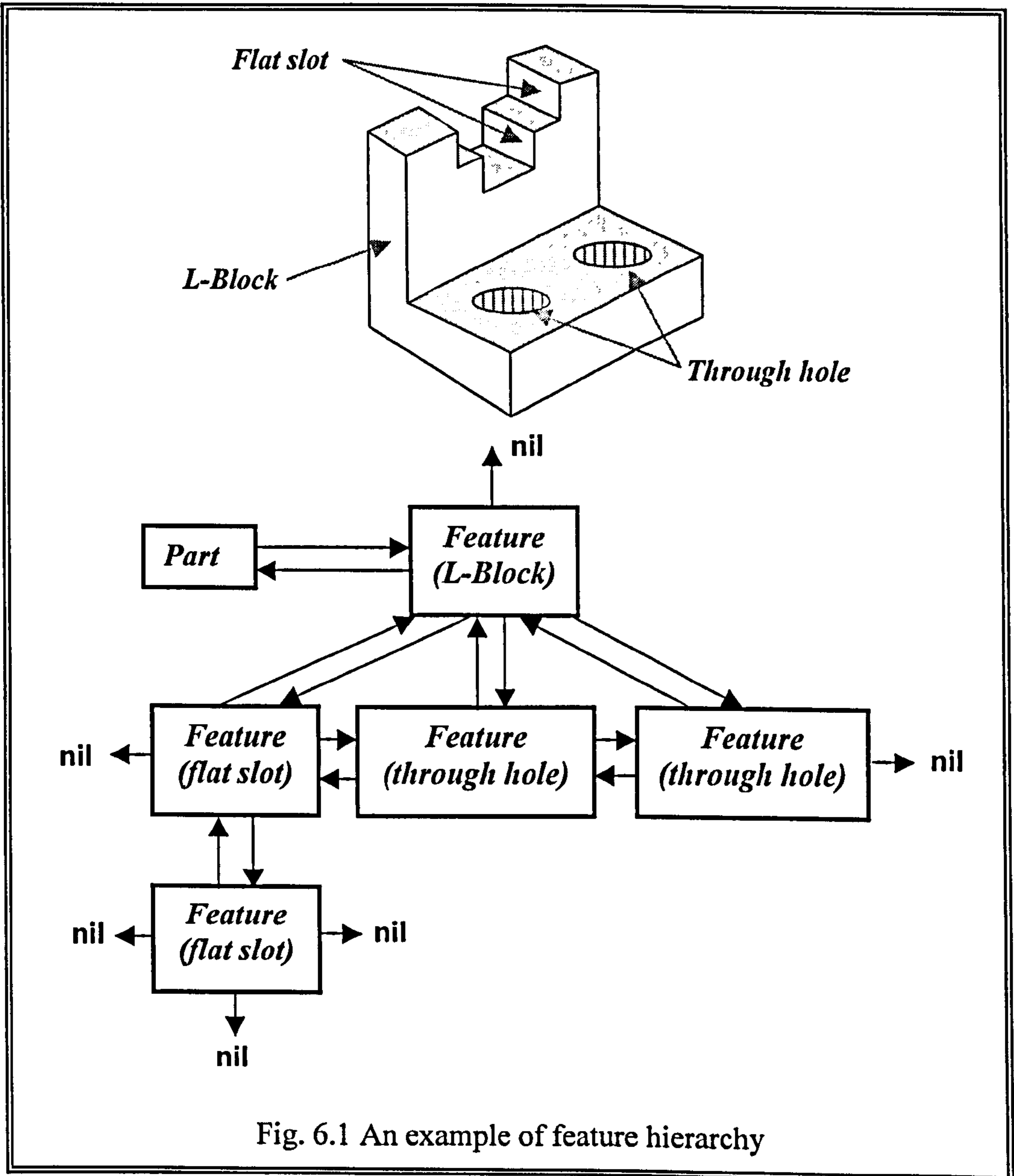


Fig. 6.1 An example of feature hierarchy

## 6.6 Feature Library

For each application, an appropriate feature library containing the generic definitions of all atomic features is required for performing the necessary tasks of the application. This is because there is no universal feature library for satisfying all CAD/CAM requirements. The number of features in the library reflects the range of the parts and applications that must be covered. There are various types of geometric features that are used in product definition. Although the number of possible features is not finite, it may be possible to categorise features into some classes that are relatively independent of the intended application domain of the features. To extract complete information of a feature from a set of entities, the generic definition of features must be clearly divided in the feature library.

Gindy, N. (1989) specified all of the features into a hierarchical tree structure according to their characteristics. At its top level this taxonomy has form features: depression, protrusions and surfaces. Each of these three categories is then subdivided depending upon the number of external access directions that are present. The resulting classes are then further divided down depending upon the boundary of the feature, which could be open or closed. The Gindy's form-feature taxonomy was adopted to be some of input nodes of the artificial neural network (ANN) module implemented in this application.

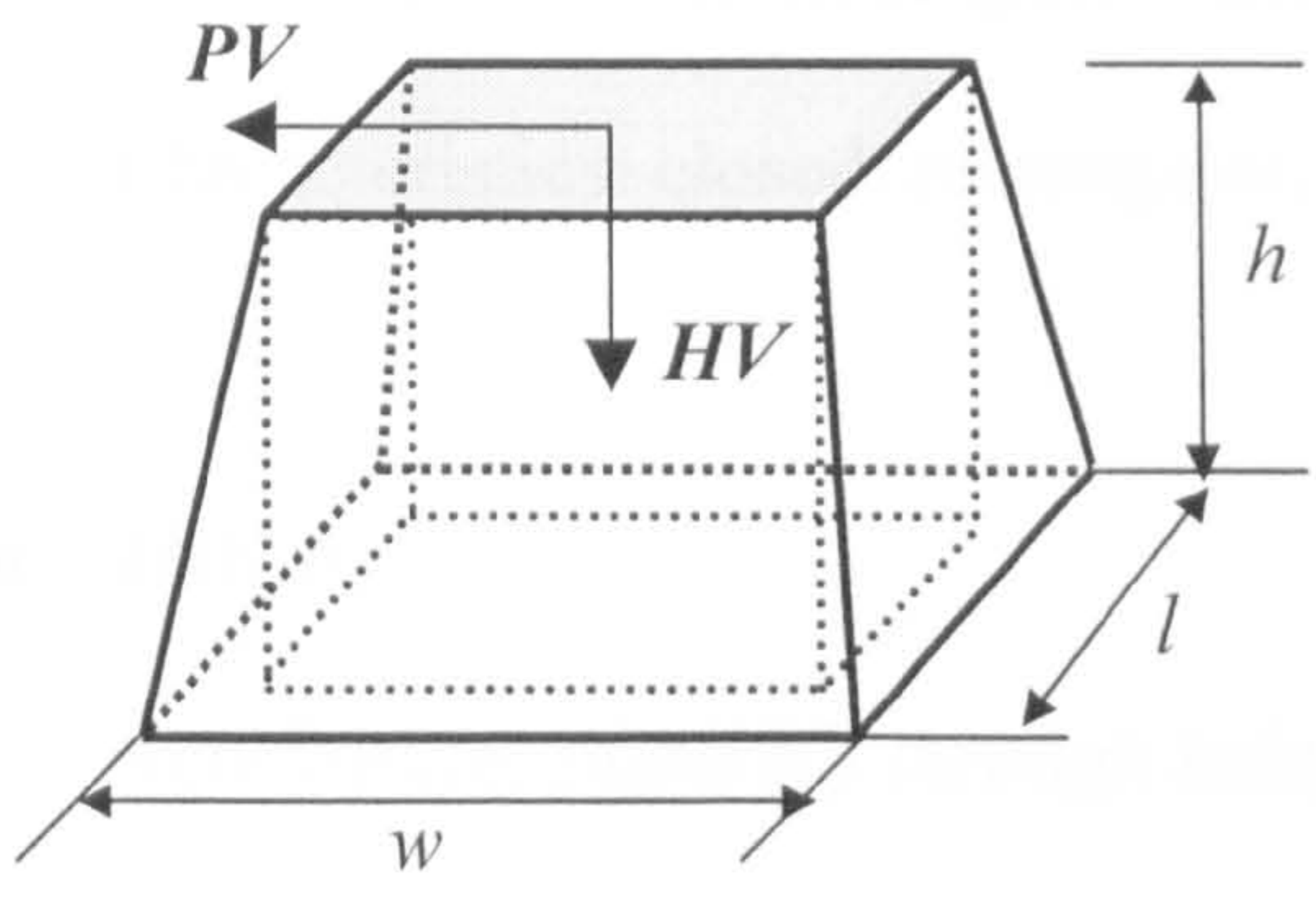
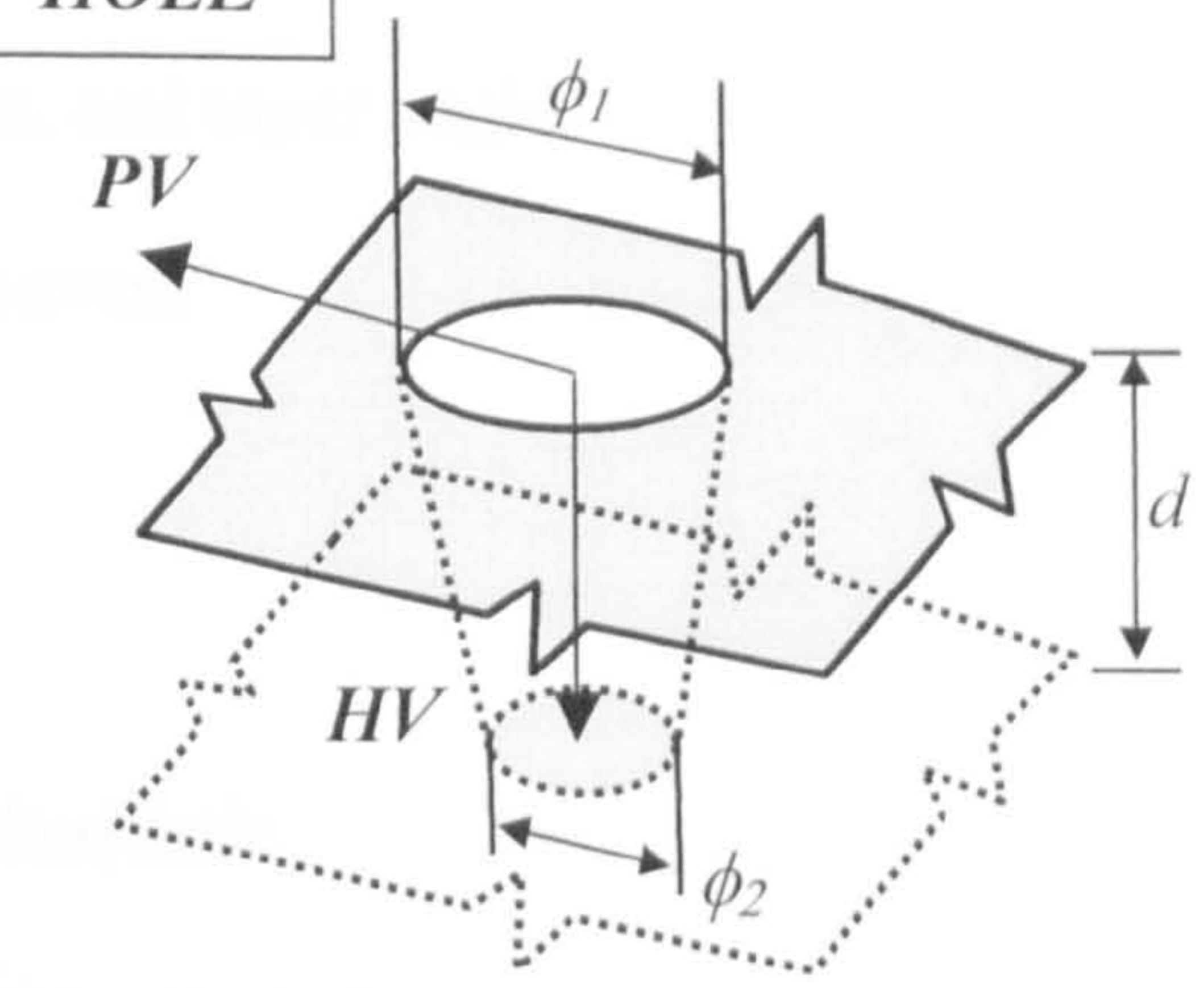
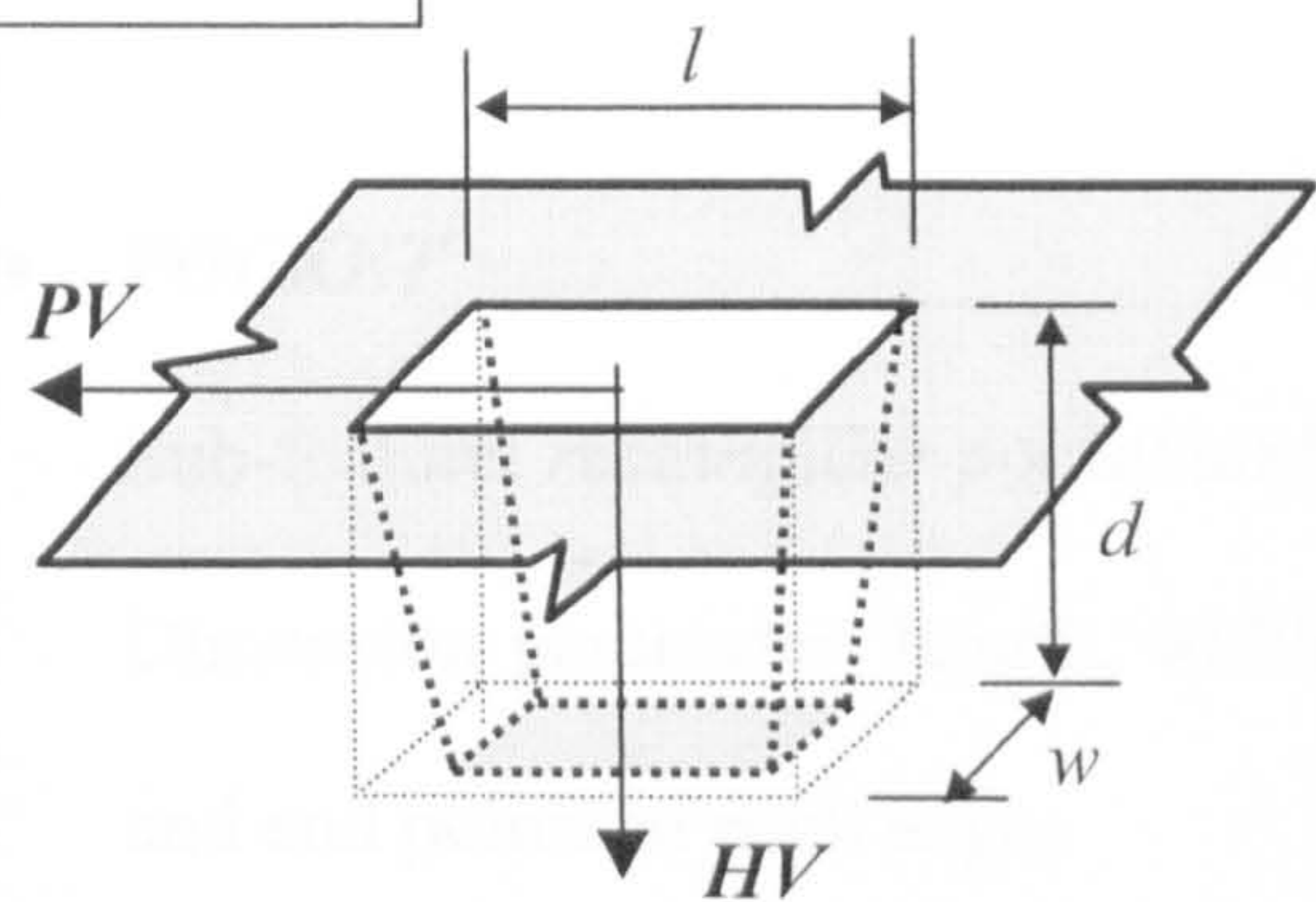
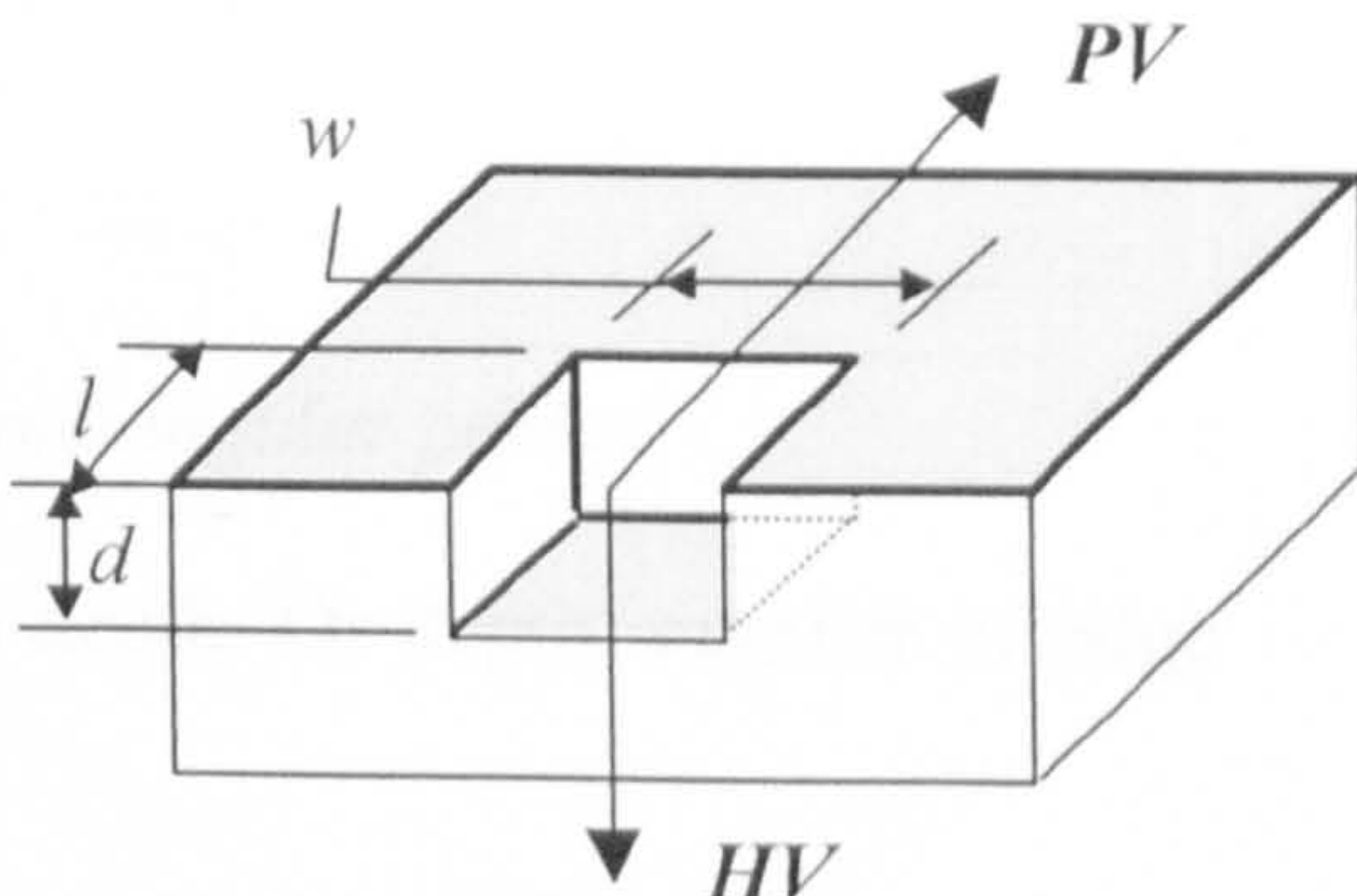
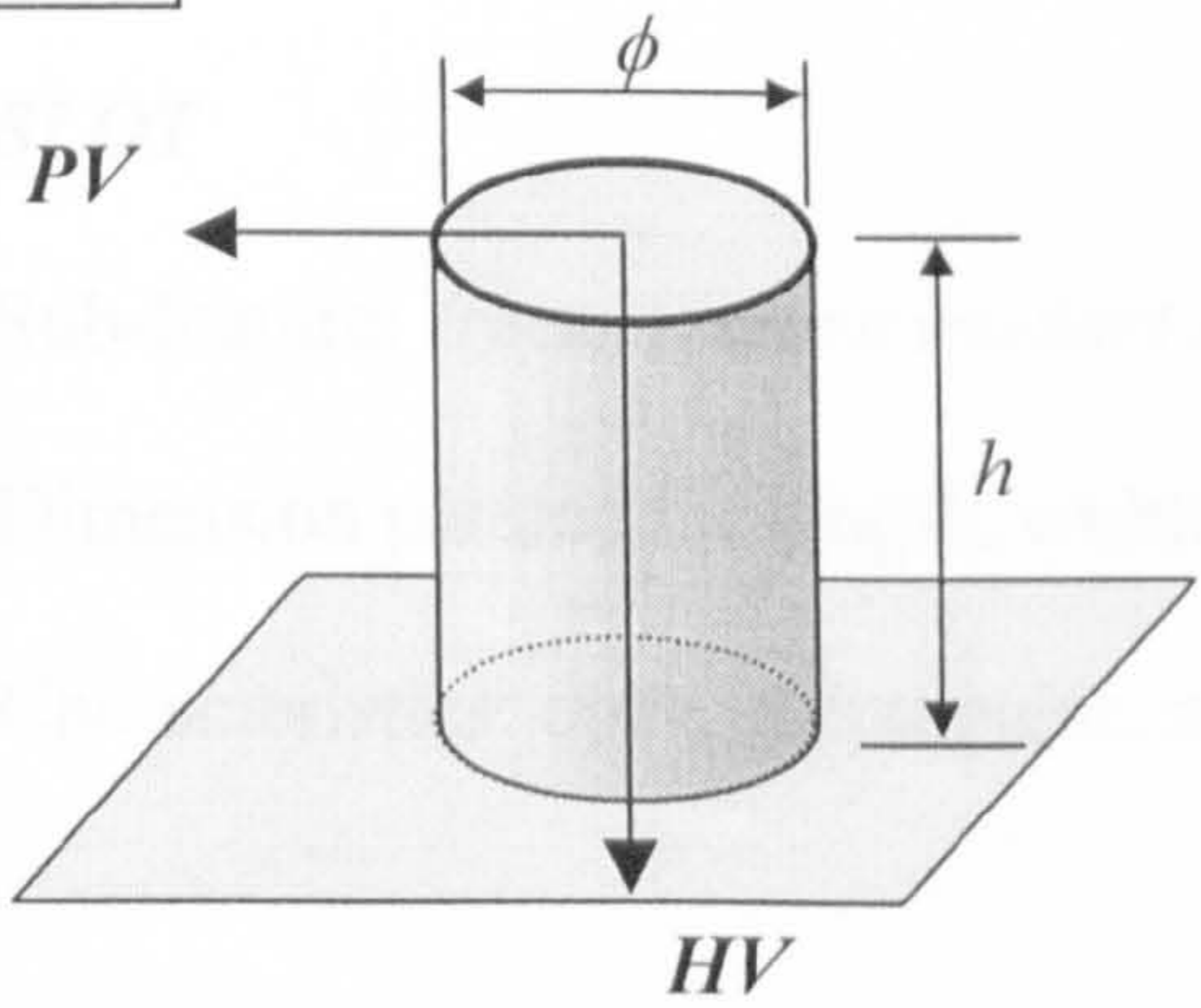
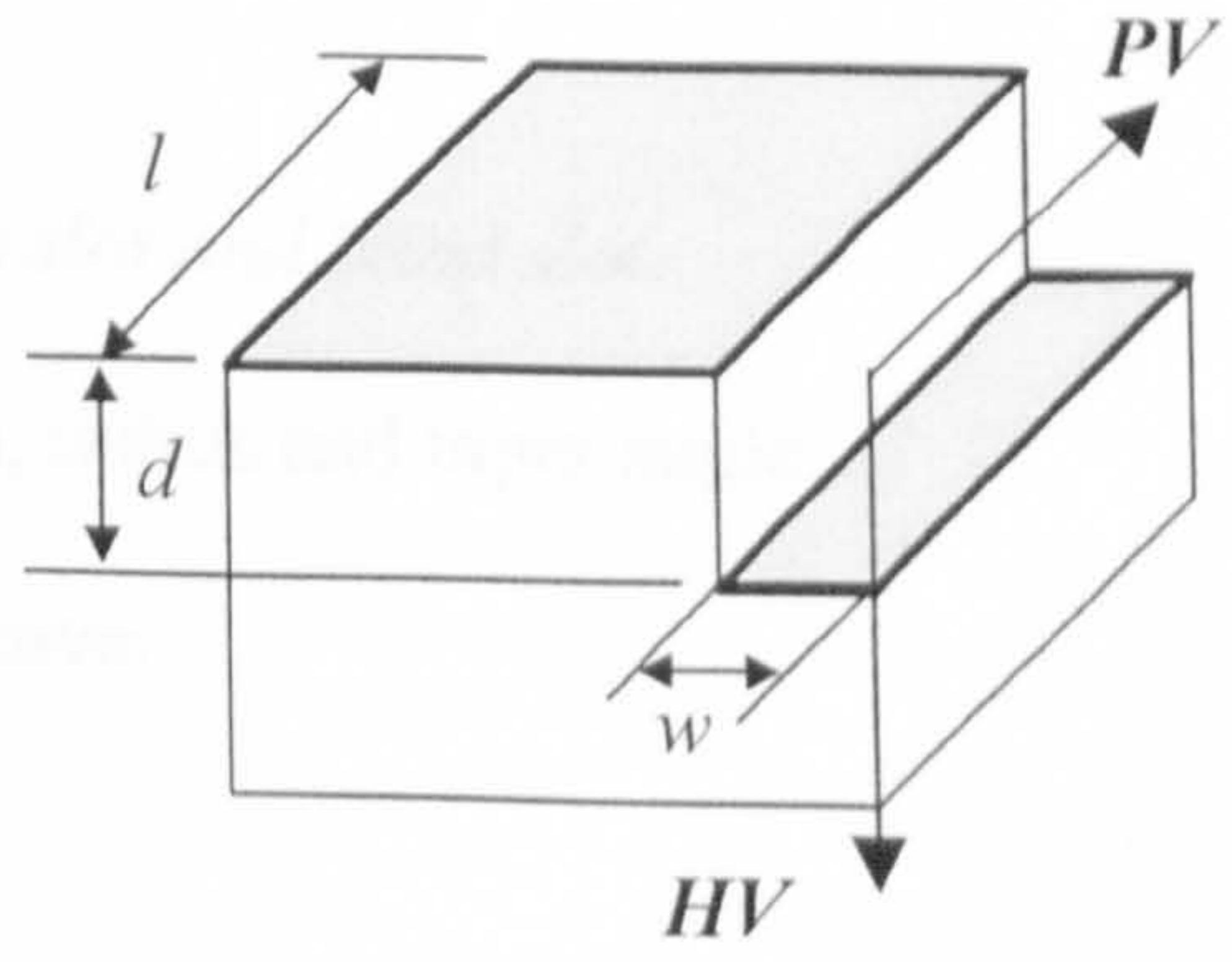
In this work, the STEP classification of volume feature taxonomy was used as the basis of the feature library. More detailed explanation about STEP data is given in the section 8.2.

This feature library is currently composed of six  $2\frac{1}{2}D$  generic features such as pocket, hole, boss, block, step, and slot, but the feature set can be expanded further.

The shape of each generic feature is defined by size dimension and orientation on the part model. Each feature can be positioned in world coordinates system (WCS) or local coordinates system (LCS), and detailed descriptions of feature adopted are as follows, as shown in Table 6.1.



Table 6.1 Generic feature taxonomy in the feature library

<p><b>BLOCK</b></p>  <p>Length, width, height, t-angle, HV &amp; PV</p>	<p><b>HOLE</b></p>  <p>Diameter, depth, t-angle, HV &amp; PV</p>
<p><b>POCKET</b></p>  <p>Length, width, depth, t-angle, HV &amp; PV</p>	<p><b>SLOT</b></p>  <p>Length, width, height, HV &amp; PV</p>
<p><b>BOSS</b></p>  <p>Diameter, height, t-angle, HV &amp; PV</p>	<p><b>STEP</b></p>  <p>Length, width, depth, HV &amp; PV</p>

- **BLOCK**

Sub-feature: *cube, general block, and wedge.*

Dimension parameter: length, width, height, and taper angle.

Characteristics: closed, rectangular, and convex.

- **HOLE**

Sub-feature: *(tapered) through hole and blind hole.*

Dimension parameter: diameter, depth, and taper angle.

Characteristics: closed, circular, and concave.

- **POCKET**

Sub-feature: *rectangular pocket and non-rectangular pocket.*

Dimension parameter: length, width, and depth. Or depth, number of edge and end points of each edge.

Characteristics: closed, rectangular, and concave.

- **SLOT**

Sub-feature: *(rectangular/circular) through slot and blind slot.*

Dimension parameter: length, width, height, radius and taper angle.

Characteristics: open, rectangular, and concave.



- **BOSS**

Sub-feature: *(tapered) boss*.

Dimension parameter: diameter, height, and taper angle.

Characteristics: closed, circular, and convex.

- **STEP**

Sub-feature: *rectangular step and non-rectangular step*.

Dimension parameter: length, width, depth, radius, and taper angle.

Characteristics: step, rectangular, and concave.

*Common parameter of all generic features: origin position and orientation vector.*



## 6.7 Feature Recognition Methods

Feature recognition can be described as the finding of features within a set of geometric entities after creation. The process of feature recognition encompasses three major tasks (Prebhakar, 1992):

- (1) *feature definition*, in which the rules for recognition are specified,
- (2) *feature recognition*, in which candidate features are identified,
- (3) *feature extraction*, in which features are extracted from the low-level data, and stored for further analysis.

Although much research work has been conducted involving automatic feature recognition, the methods are intended to work with existing CAD models. A geometry model is created first, then a computer program processes the database of the model to extract engineering features. The method starts with a conventional volumetric representation of part geometry, then derive an alternate representation in terms of features, and then use this information as an aid in process planning. However, even the B-rep based feature identification schemes have been developed mainly for research programs. Only simple examples have been used, and in some cases there are significant limitations on the sorts of geometry that may be examined (for instance models with planar faces only). The recognition rules or algorithms may also be confused by such things as intersecting features – for example intersecting pockets.

There have been several methods for feature recognition: graph-based, set-based, rule-based and syntactic-based recognition.

### 6.7.1 Graph-based recognition

The graph based recognition method has been widely used for feature recognition from B-rep CAD models. This is because B-rep based solid models have usually graph-based structures which characteristics are based on the number of entities, topologic type, connectivity and adjacency (Joshi, 1988). A complete description of adjacency relationships between faces, edges, and vertices is used to feature extractions. This graph is also called as attributed adjacency graph (AAG). The recognition processes are carried out to look for *cut faces* that is a node in the graph which would cause the graph to fall into at least two parts as shown in Figure 6.2 (Stephen, 1993). These parts are then analysed recursively until no more *cut faces* exist.

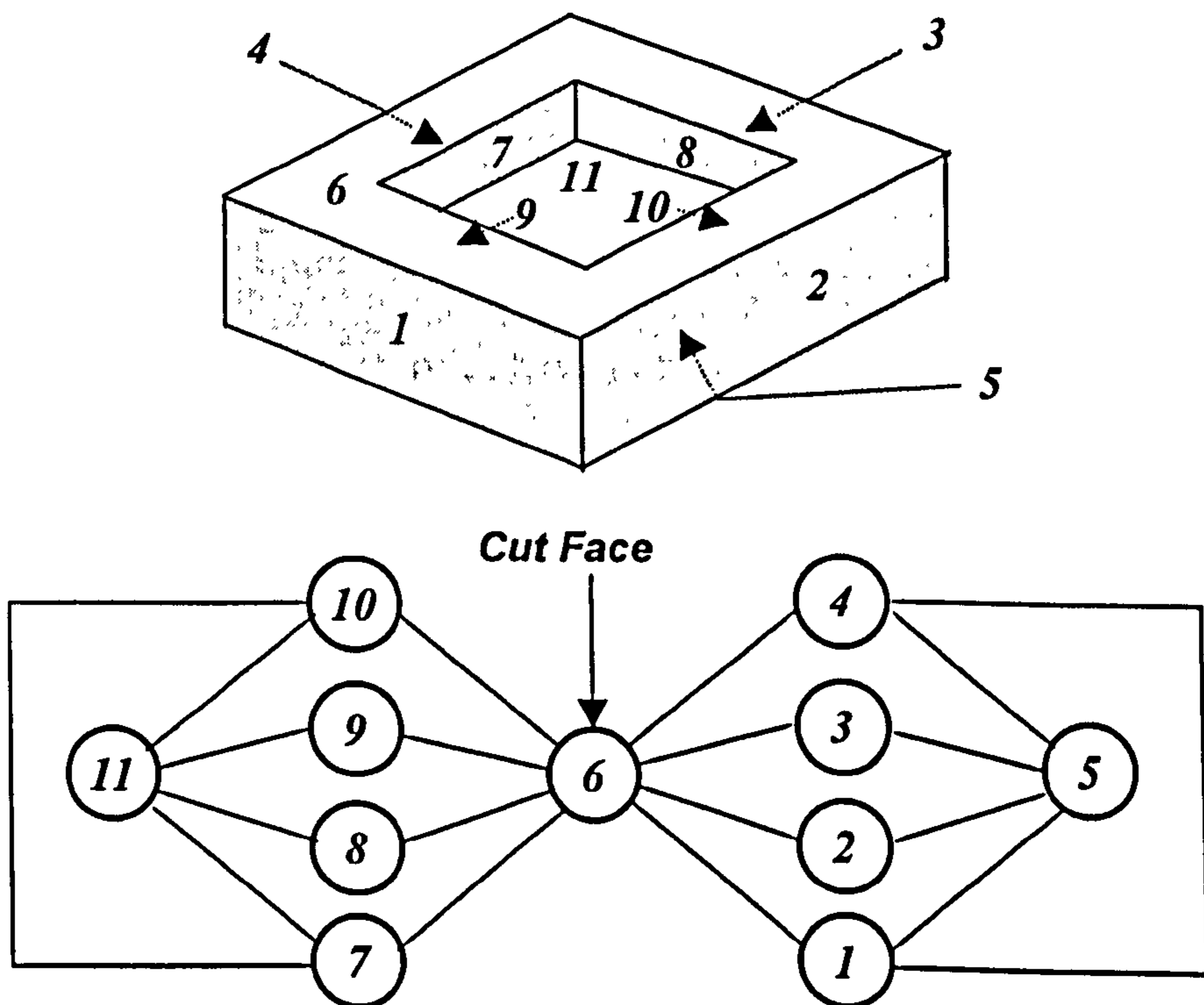


Fig. 6.2 A solid model with *face adjacency graph* (Stephen, 1993)

### 6.7.2 Set-based recognition

Set-based recognition yields an unique series expansion of the model in terms of convex volumes with alternating signs. These can then be assembled using the alternating sum of volumes to highlight the model's constituent primitives (Woo, 1982). This technique is usually applied to CSG models to extract features. Kim, Y. (1993) presents a technique that can convert positive features to negative machining features. The technique is based on combining positive features with a appropriately chosen negative feature; as a result equivalent negative components are extracted.

### 6.7.3 Rule-based recognition

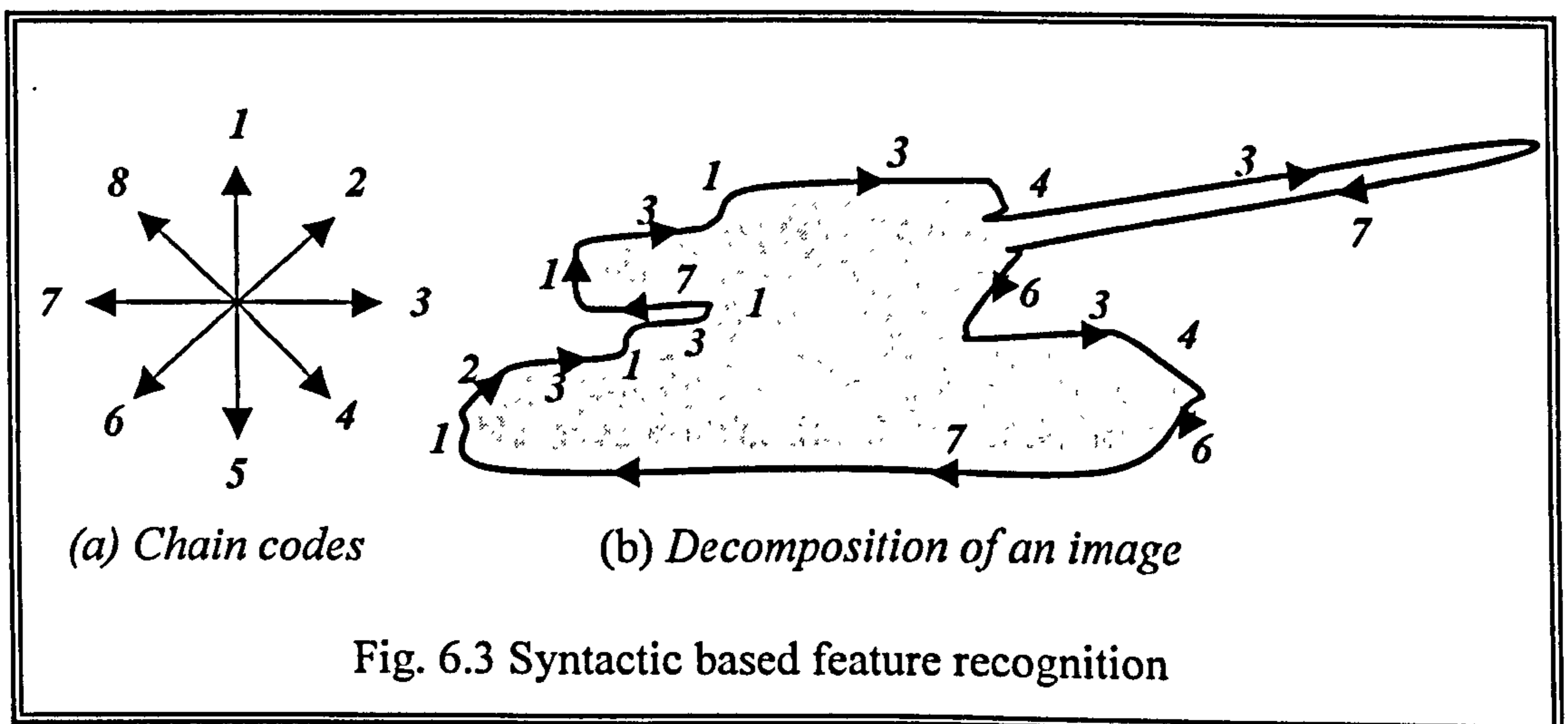
Donaldson, I. (1993) adopted a rule-based approach, which recognise features on the basis of rules that investigate the underlying model. Features are defined by both general features (depression, protrusion, passage) and specific features (rectangular pocket, blind hole, through slot, etc.). The geometric and topological conditions of such rules have to be satisfied in order for a rule to be satisfied.

Rule-based systems cannot generalise from past experience to handle new, unforeseen cases. Rules are non-unique to a feature combination and rules cannot be devised for every conceivable feature combination. Rule-based systems have limited ability to learn or to improve their behaviour dynamically (Devireddy, 1999).



### 6.7.4 Syntactic-based recognition

Syntactic-based recognition was applied to features by Choi *et al* (1984) and Jain *et al* (1998). Basic to the approach is the decomposition of patterns into sub-patterns or chain codes, so patterns are described by a series of typically straight, circular, or other curved line segments. The decomposition process continues until the simplest patterns are derived. Once the sub-patterns are determined, the next step is to define rules that describe relations among these sub-patterns, so that valid composition of sub-patterns is guaranteed. Fig. 6.3 (b) shows such a decomposition of an image in terms of the chain codes defined in Fig. 6.3 (a).



By tracking the boundary of the image in a clockwise direction, it may be possible to encode these chain codes in the form of a string of qualifier. Thus, the image can be represented by the string “3437634671231317131” from the top.

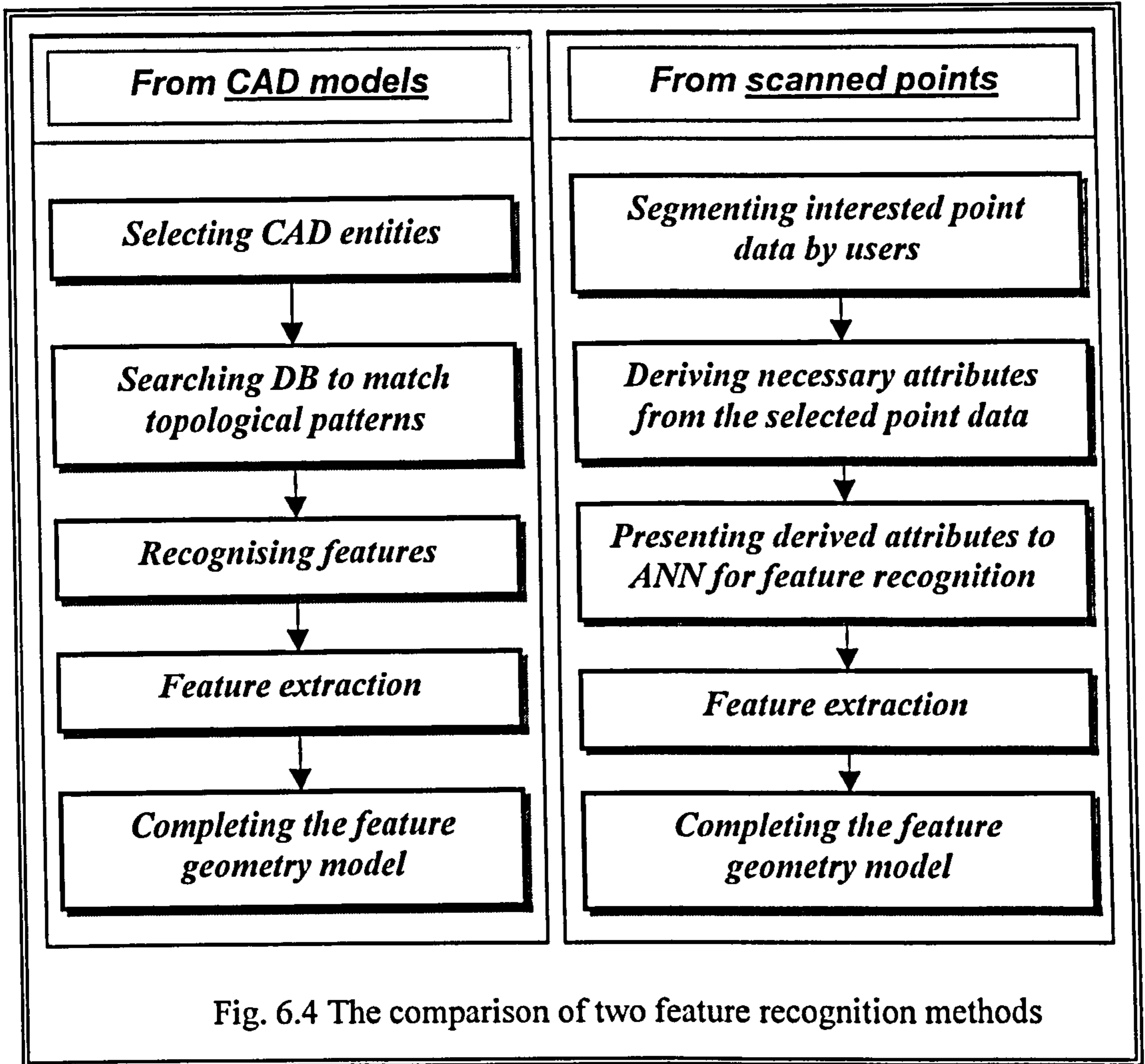
## 6.8 The proposed Feature Recognition Method

In this research, a learning process based on artificial neural network (ANN) algorithms was adopted. This is because the background geometric models are scanned point data that have no topological information except pure 3D coordinates. ANN systems are a special case of rule-based systems that allow probabilistic recognition of partial features that often result from feature interactions (Shah, 1995). The algorithm of syntactic based recognition was also employed to generate geometric attributes for the ANN module. The geometric patterns of scanned point data are typically described by a series of line segments. Simple strings can be concatenated to give full patterns. The result is a set of features matched with the part. The output features can completely define the geometric and topological representation of the part.

The advantages of ANNs over conventional machine learning methodologies are (Devireddy, 1999):

1. parallel consideration of multiple constraints;
2. capability for continued learning throughout the life of the system;
3. graceful degradation of the performance; and
4. ability to learn arbitrary mappings between input and output domains.

The flow charts comparing the two feature recognition methods are schematically depicted in Fig. 6.4. A more detailed description of the feature recognition methodology implemented is fully discussed in chapter 7.





# Chapter 7

## Feature Recognition using Artificial Neural Networks

Feature recognition is the process of identification of expected patterns of geometry, corresponding to a particular engineering functionality in parts of an object (Tyan, 1998). Feature recognition algorithms working directly on 3D model inputs are complex even when the features are simple. Many different techniques have been attempted to automatically derive information on features and geometry from solid models. As discussed in previous chapter, techniques used in feature recognition research include syntactic pattern recognition, geometry decomposition, expert systems, graph-based methods, convex hull method, and so forth. However, those conventional procedural methods are usually very time-consuming and not suitable for feature recognition directly from coarse 3D scanned data. Due to the diversity of scanned points according

to input part shapes, appropriate information from a scanned data set is unlikely to be derived.

Since the author found no published evidence of this feature recognition approach being evaluated previously, the proposed approach is unique. An artificial neural network (ANN) is employed to automatically recognise features directly from a set of point clouds since ANNs are best at identifying patterns or trends in sophisticated data (Freeman, 1992). ANNs differ from most other classes of artificial intelligence (AI) tools in that the networks do not require any clear-cut rules or knowledge to perform tasks. In essence ANNs possess the ability to make generalisations and perform reasonably on input patterns that have never previously been presented with. They resemble the human brain in the sense that training rather than programming is used to acquire experiential knowledge. Knowledge regarding feature recognition is encoded in the set of weights during training process.

ANNs require parallel input patterns. In this research, a group of 2.5D information extracted from 3D scanned points was the given input to the ANN. Each feature in the feature library was uniquely described by three geometric attributes: *chain codes*, *convex/concave*, *circular/rectangular* and *open/closed attribute*. Identifying each feature requires the determination of these attributes. The primary features in the feature library are classified based on these attributes. If these attributes are determined, a given subpart may be uniquely

identified as a valid library feature. New and robust algorithms have been developed for determining these three attributes for each of the features.

The proposed feature recognition process is summarised in Fig. 7.1, and its detailed steps are then discussed.

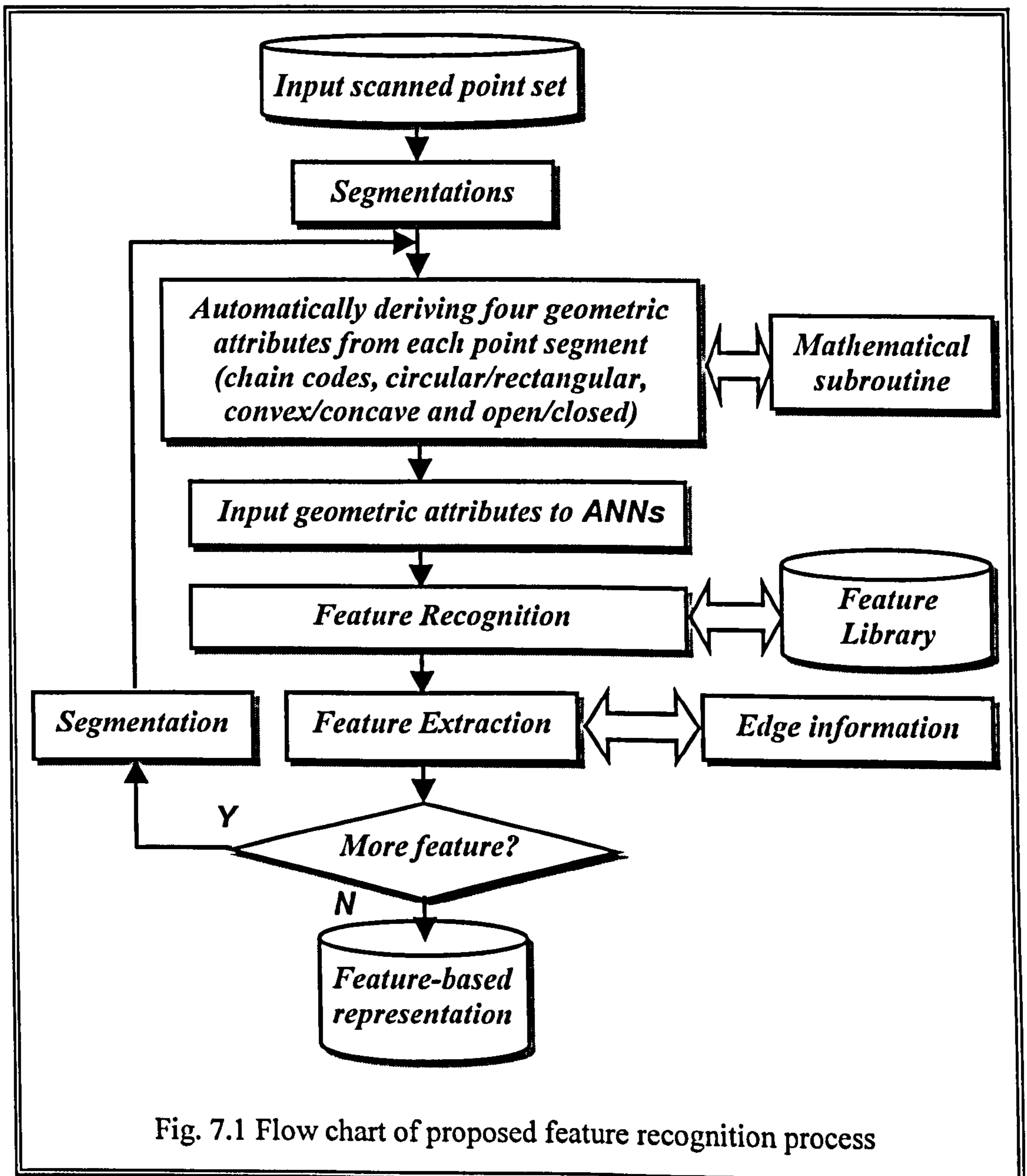


Fig. 7.1 Flow chart of proposed feature recognition process



Once the user selects an appropriate portion of scanned point data on the displayed image using the mouse (segmentation), the application then analyse the selected point data to derive geometric attributes such as chain codes, circular/rectangular, convex/concave and open/closed using corresponding modules. These attributes are presented to the input layer of the ANN module. The ANN module runs the network to recognise a feature corresponding to the input attributes, using its previously trained weights. The output of the ANN module is equivalent to the type of recognised feature.

The aim of this chapter is to provide a new concept of feature recognition algorithms directly from a scanned point set. The remainder of this chapter is organised as follows. The first section (7.1) briefly describes the core algorithm of ANNs. The next section (7.2) reviews the previous research works regarding ANN-based feature recognition techniques. Then the following two sections (7.3 and 7.4) discuss the steps of the proposed feature recognition process and the ANN structure implemented in this research. Finally, the algorithms regarding deriving four geometric attributes are precisely discussed in section 7.5.

## 7.1 Introduction to ANNs

### 7.1.1 A simple neuron

ANNs consist of three or four *layers*. Layers are composed of a large number of highly interconnected processing units, called *neurons*<sup>1</sup>, which can work in parallel to solve a problem, as shown in Fig. 7.2.

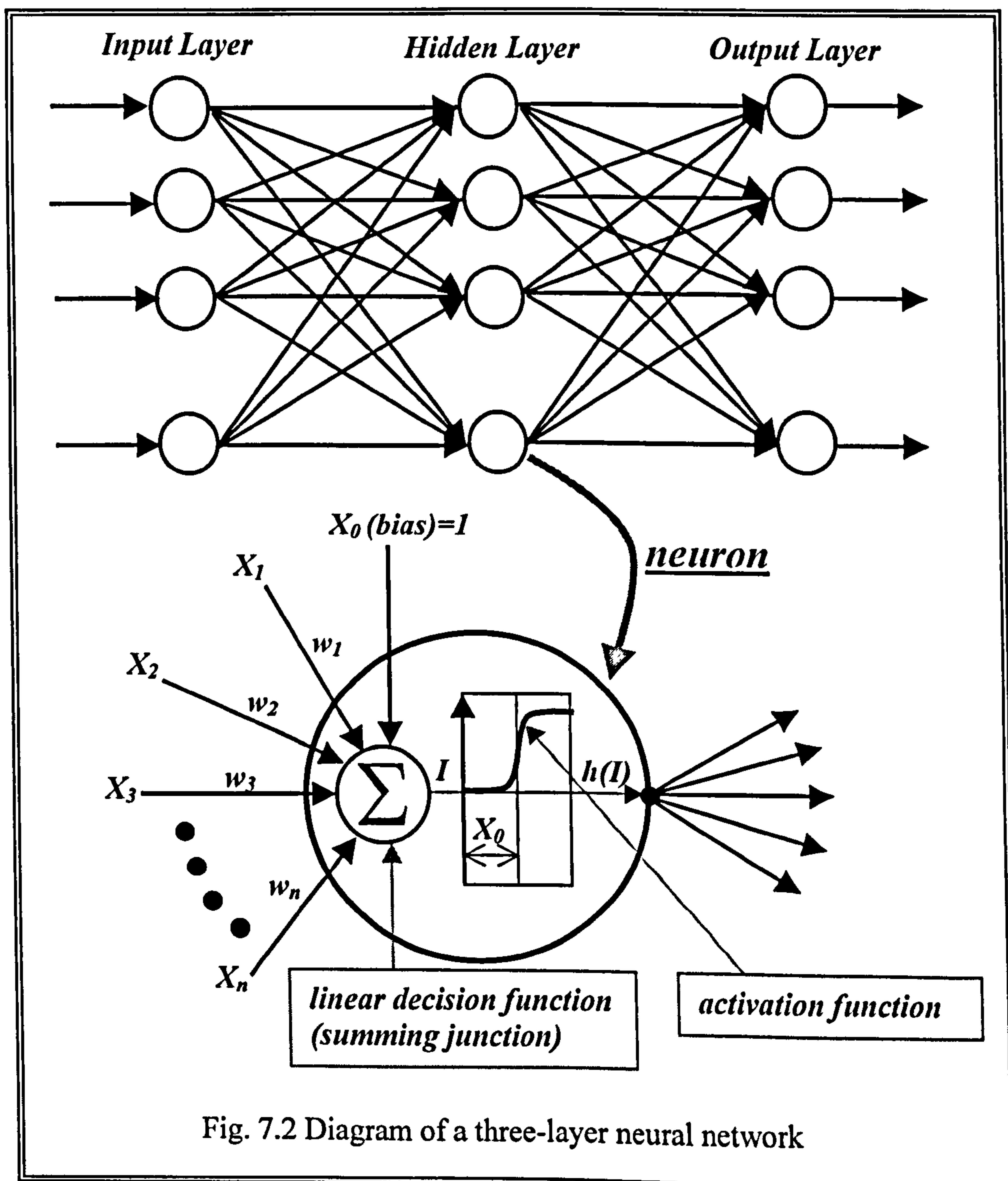


Fig. 7.2 Diagram of a three-layer neural network

<sup>1</sup> The human nervous system consists of approximately  $10^{11}$  neurons.

Each circle in Fig. 7.2 represents a neuron. A neuron is an information process device inspired by the operation of biological nervous systems (Kohonen, 1988). It receives input stimuli from the neurons of previous layer, and it sends its internal state (activation) as a signal to other neurons in the next layer. A neuron can have several inputs, but has only one output. This output is connected to the inputs of several other neurons.

A neuron contains its bias (*threshold*), an *activation function*, and the weights associated with its inputs. Here,  $I$  and  $h(I)$  is the input and the output of the neuron, respectively.  $X_i$  is the  $i^{\text{th}}$  input value,  $w_i$  is the weight associated input  $i$  and  $X_0$  is the bias. A bias element is considered to be an input neuron with constant activation value of 1.0, playing the role of an activation threshold.

Each neuron performs a very simple arithmetic operation, i.e. it calculates the weighted sum of its inputs, subtract its threshold from the sum, and passes the result through its activation function; that is,

$$I = \sum_{i=1}^n w_i X_i + X_0 \quad (7.1)$$

which is a linear decision function with respect to the components of the pattern vectors. The function that maps the output of the summing junction into the final output of the device is called the *activation function* or *transfer function*. It may be a linear or a nonlinear function of the net input  $I$ . A unipolar or a bipolar sigmoid function is widely used as the activation function. The activation function is required to have several important characteristics: continuous,

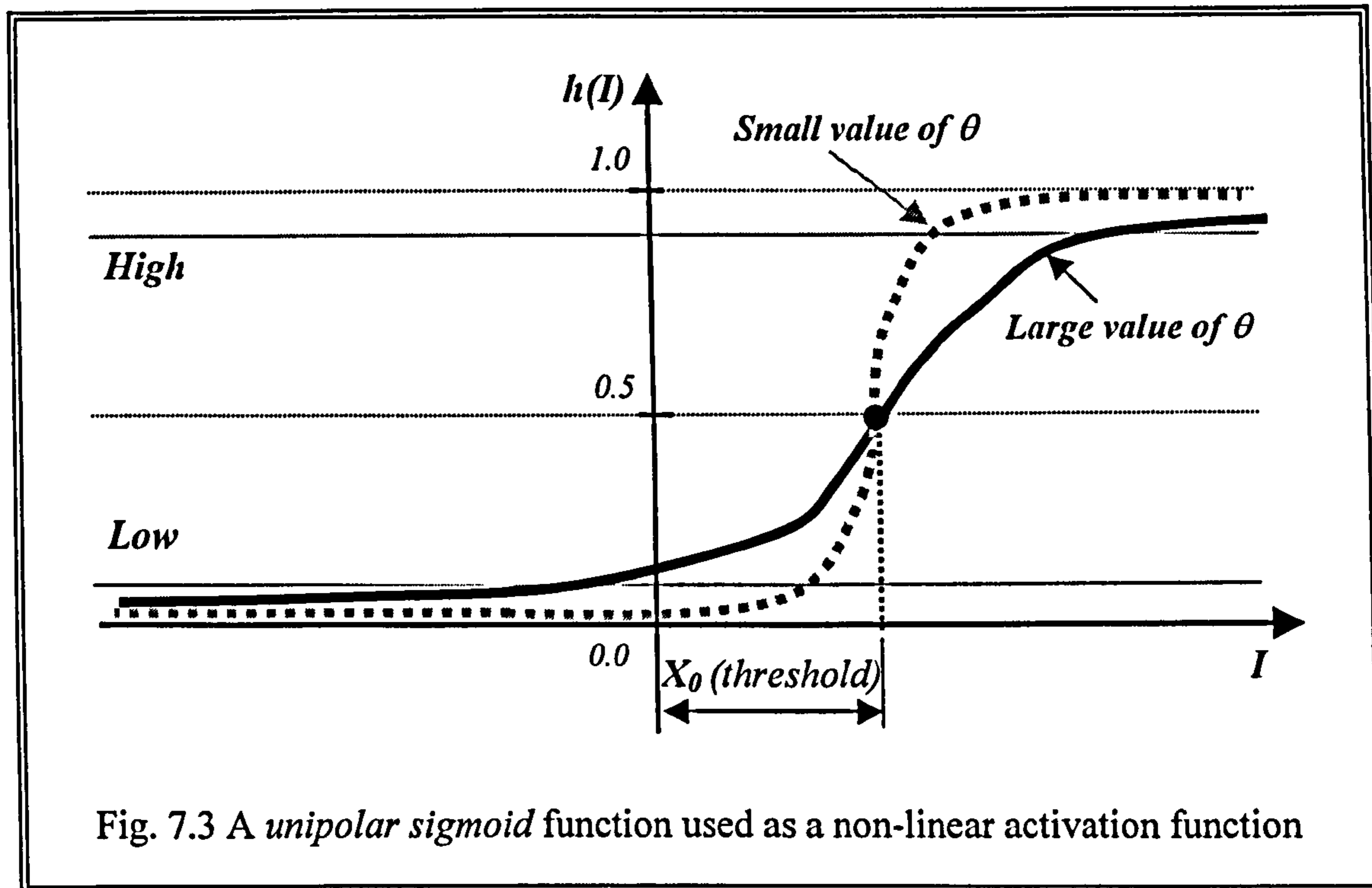


differentiable, and monotonically non-decreasing (Freeman, 1992). A sigmoid function describes how a neuron reacts to the external signal and generates its initial excitability value. A unipolar sigmoid function was used in this application.

The activation (output) of a neuron is the value of the sigmoid function of the weighted sum of the neuron, that is:

$$h(I) = \frac{1}{1 + e^{-(input)}} = \frac{1}{1 + e^{-\frac{I}{\theta}}} = \frac{1}{1 + \exp\left[-\left(\sum_{i=1}^n w_i X_i + X_0\right) / \theta\right]} \quad (7.2)$$

where  $X_0$  is an threshold,  $n$  denotes the number of neurons in the previous layer,  $X_i$  is the input to the neuron from the neuron  $i$  in the previous layer, and  $\theta$  is a arbitrary constant that controls the shape of the sigmoid function. The smaller the value of  $\theta$ , the steeper the curve shape as shown in Fig. 7.3.



Normally, in an ANN, the value of  $\theta$  is often 1.0. The figure shows a shifted curve to the right by the distance  $X_0$  because the activation and the weight of the bias are supposed to be 1.0 and positive, respectively. Thus when a sigmoid function is used, the system outputs a high reading for any value of  $I$  greater than  $X_0$ . Similarly, the system outputs a low reading for any value of  $I$  less than  $X_0$ . The unipolar sigmoid activation function is always positive, and the output varies continuously but not linearly as the input changes. It can reach its limiting values of 0.0 and 1.0 only if the input to the activation element is infinitely negative or positive, respectively. For this reason, values near 0.0 and 1.0 (i.e., 0.05 and 0.95) define high and low values at the output of the neurons. Output signals travel along other weighted links to connected neurons.

A neuron individually functions rather slowly, but collectively performs tasks at amazing speed. Each connection between neurons is associated with a weight

which is adjusted by training data. The meaning of the connections is interpreted as the relations between the neurons. The number of connections among a network is so large that it provides the network with powerful capabilities such as logical derivation, object perception in natural scenes, and so forth (Freeman, 1992).

### 7.1.2 Training by a back-propagation (BP) algorithm

Training or learning occurs by modifying the weights of an ANN to obtain the desired output with a set of training data pairs. Among the many different kinds of learning rules, the most widely used learning mechanism for multi-layered perceptrons is BP algorithm. A BP is a generalisation of a least-mean-squares method, called a *steepest-descent technique*. It performs a gradient descent within the solution's vector space toward a global minimum along the steepest vector of the error surface. Its learning procedure is based on gradient search with the least-sum-square criterion. Most learning rules have built-in mathematical terms to assist the process, which control speed and momentum of the training. The training of a multi-layered neural network using *BP* is a non-linear optimisation problem, so an iterative process is usually adopted to solve the problem.

The learning process of a BP takes places in two phases. In the forward phase, the learning process is carried out by presenting a sequence of input training patterns with an associated desired output set. The output of each neuron in each layer and the errors between the actual outputs are computed. In the backward phase, weights are modified by back-propagated errors that occurred in each



layer of the network. After many repetitive presentations (*epochs*) of the training patterns, the output of the neural network converges to the desired outputs. This process is critical to the performance of *ANNs*. The detailed BP training procedures are given in *Appendix 4*.

### **7.1.3 What are the differences between *ANNs* and conventional procedural computing?**

A conventional procedural computation is processed sequentially and logically, and the state of a given variable can be obtained from a set of instructions and tracked from one operation to another in order to solve a problem. The specific and well-arranged steps need to be given to the computer. The computational steps are deterministic. However, this procedural computation is not suitable for feature recognition in RE, since it may not cover all possible patterns of various input scanned points.

In comparison, *ANNs* take a different approach to problem solving than that of conventional methods. *ANNs* can capture many kinds of relationships or discovering regularities within a set of input patterns that are difficult to describe adequately with conventional approaches. *ANNs* do not execute programmed instruction; they respond in parallel to the input patterns presented to them. *ANNs* are not necessarily deterministic, and there are many simple neurons which generally take the weighted sum of their input from other neurons. There are also no separate memory addresses for storing data. Instead, information is contained in the overall activation 'state' of the network, which is quite literally more than the sum of its individual components. To respond

correctly to given input patterns, it is necessary to train the network. Once a neural network is fully trained, it can be applied to problems where the relationship between input and output may be quite dynamic or non-linear.

However, there are also several limitations to ANN analysis. Firstly, the user is unlikely to know what is going on to the network during training. Apart from providing it with a initial random numbers, the user has no other role than to provide it with input and watch it train and await the output. The final output of this training activity is a trained network that provide no equations and coefficients defining a relationship between sampled input and sampled output.

Secondly, if the ANN is being trained on a standard serial machine such as PC or rather less powerful workstation, training may take time. Moreover, if it involves with a large number of training data in a very large network, it would take even more time. However, the speed of most current machines is so fast that this is typically not much of a critical issue.

Finally, the robustness of ANNs strongly depends on the nature of the application and the reliability of the training data. This means that the user needs to have got sufficient experience to provide training data to the ANN. The training data must be selected carefully otherwise useful time is wasted or even worse the network would not be trained at all.

## 7.2 Previous Work in Feature Recognition using ANN

A literature survey reveals that little work has been done on feature recognition using ANNs other than following works: Prebhakar, 1992; Wu, 1994; Nezis, 1997; Wu, 1997; Chen, 1998. Prebhakar (1992) firstly employed ANN techniques to recognise form-features from B-rep solid models of parts. In their work, the adjacency matrix describing a part was used as the input to the network. However, no training step was addressed in the sense of repeated presentations of input training patterns. Wu, M. *et al* (1994) used topological information to recognise 2D part with examining the 'skeleton' of parts. Gu, P. (1995b) applied ANN approach to the reconstruction of a damaged or worn free form surface. They used a series of points from a mathematically known surface to train the network. Another approach in feature recognition that uses ANNs is the work of Nezis *et al* (1997). They used a set of heuristics in the step that breaks compound features into simple ones. Wu, P. *et al* (1997) presented a learning mechanism for parts recognition in an intelligent assembly system with a variable learning rate and a variable searching direction. More recently, Chen, Y. *et al* (1998) proposed a feature recognition technology for recognising sheet metal parts created in a CAD system. However, their method was limited to recognise a feature with less than five edges. Many of the commercially available systems for the RE of mechanical parts using automatically acquired 3D position data use rather unsophisticated geometric models (Thompson, 1995).



### 7.3 The Proposed Feature Recognition Procedure

Physical features in an engineering set possess many attributes or patterns that can be used to distinguish between one or more different subsets. A pattern is a vector template of attributes for a feature. A set of patterns is associated with each individual feature. In feature recognition, portions of the CAD models are compared to predefined generic feature in order to recognise feature attributes that match the predefined feature patterns. If a feature in a part satisfies some predefined relationships and possesses specific patterns, then the feature can be recognised (Prebhakar, 1992). This is easily explained with an example. If a circular face is found in the top and bottom plane of the part and the section profile curve is *convex*, then the part is likely to be a BOSS feature.

In this research, feature recognition was defined as the grouping or partitioning of the set of points on the segment of the part such that the point set represents a feature. All information about feature patterns was stored in the ANN as a weight value. The major steps for solving the feature recognition problem proposed in this research are as follows:

- Step 1 (Training): Construct an ANN, and train it with adequate and sufficient training data corresponding to each feature type in the feature library. Once the ANN is fully trained, it does not need further training process until features in the feature library are added or removed.



- Step 2 (Segmentation): Using a mouse, select a portion of scanned points on the displayed image that can be transformed to a single feature. Previously detected edge points are usually referred.
- Step 3 (Deriving Geometric Attributes): The selected scanned data are internally analysed, and geometric attributes such as *chain codes*, *convex/concave*, *rectangular/circular* and *open/closed attribute* are automatically derived. The whole processes for deriving geometric attributes are discussed in section 7.5.
- Step 4 (Feeding): Feed the geometric attributes derived in *Step 3* into each input neuron of the ANN.
- Step 5 (Recognition): Run the ANN. If one of output neurons is active, extract the feature type associated with the active neuron. Figure 7.4 shows the shell environment of the developed application after feature recognition.

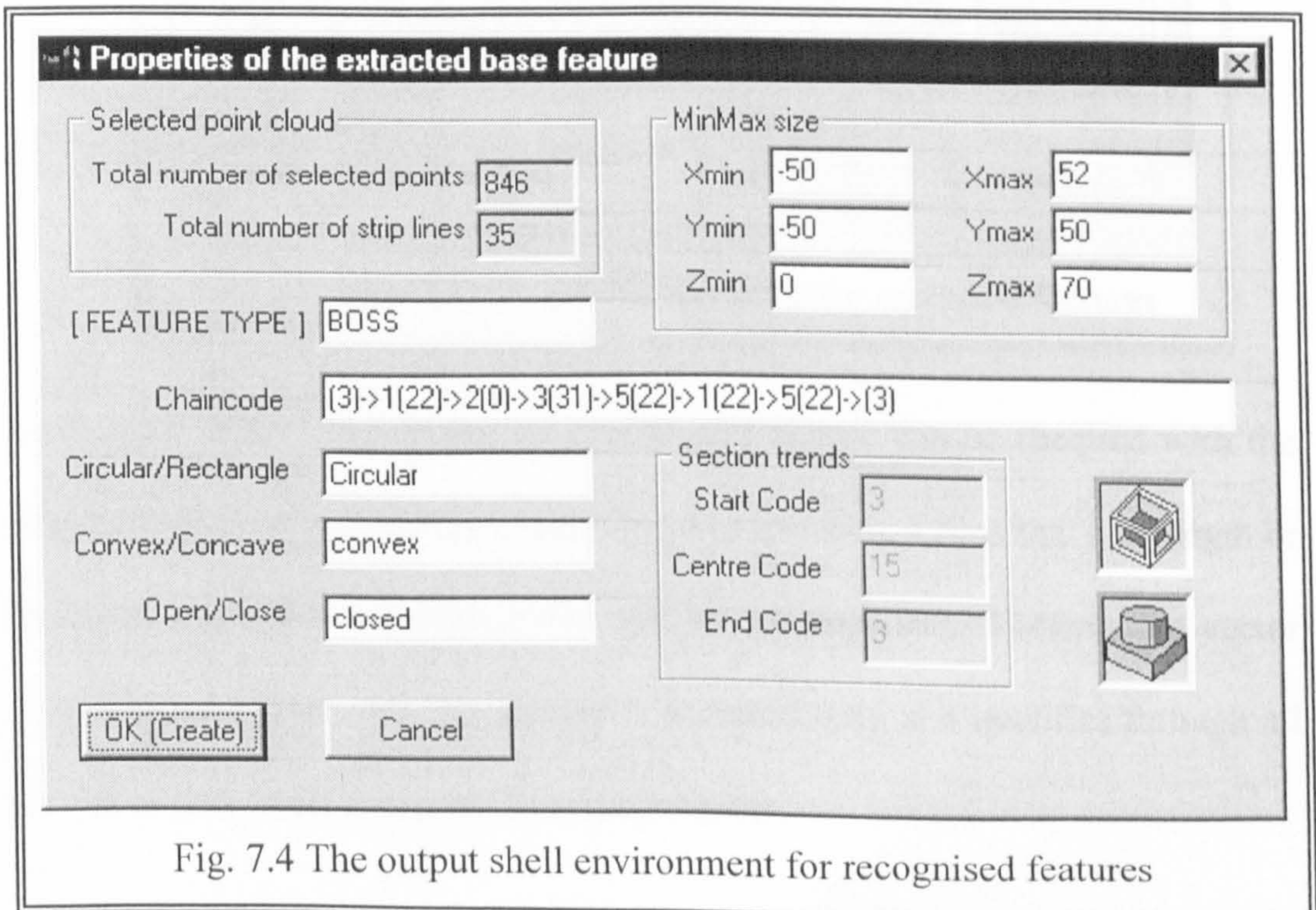
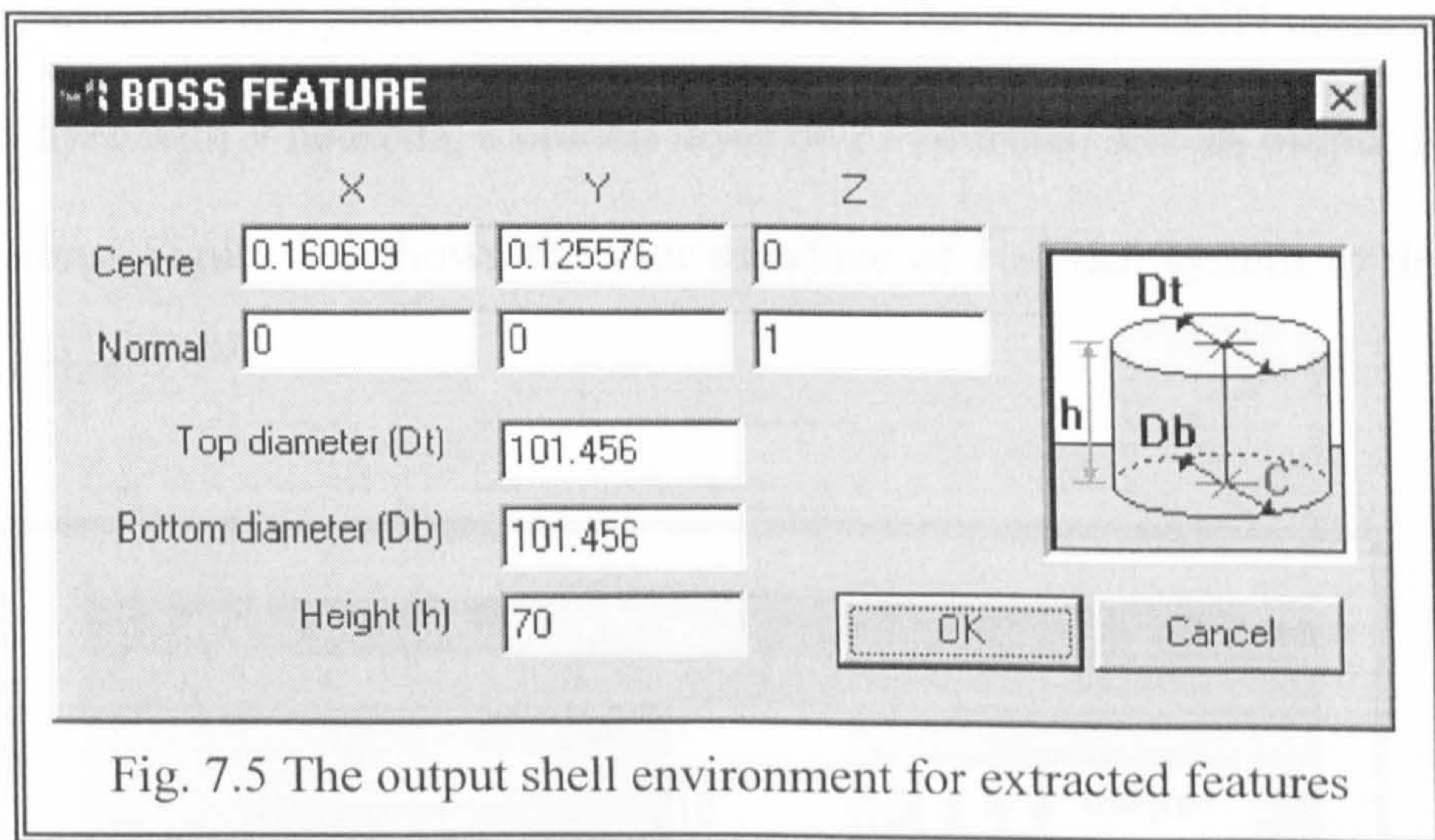


Fig. 7.4 The output shell environment for recognised features



The output of the shell give information about the recognised feature type, its derived input attributes, the number of selected points corresponding to the feature and maximum and minimum size in *Cartesian* space.

- Step 6 (Extraction): Based on the edge points of the recognised feature, its size and orientation are then reconstructed. In the Fig. 7.5, in case of a boss feature, the extracted geometric parameters are centre coordinates, orientation vector, top radius, bottom radius and length. In addition, the geometric parameters of recognised features can be easily modified by the user under the *parametric design* paradigm.



At the same time, the validity of an extracted feature can be checked with the help of the predefined feature constraints. For instance, a boss that has length or diameter less than  $0.0$  is impractical, and the  $z$ -component of orientation vector must be greater than  $0.0$ . The feature is accepted only if it qualifies through all the constraints that are imposed for each feature.



## 7.4 The ANN in this Research

### 7.4.1 The ANN structure

Many different ANN architectures have been developed (Wang, 1993). They differ in the types of propagation and activation functions, how units are interconnected, and how learning is implemented. The type of paradigms depends on the characteristics of the task to be performed: recognition, prediction or classification.

In this research, a general three-layer neural network was employed since it is sufficient to solve any problem (Freeman, 1992). The present ANN consists of an input layer with 9 neurons, a hidden layer of 15 neurons, and an output layer of 6 neurons. Figure 7.6 shows the user interface of ReTool system to define the ANN size.

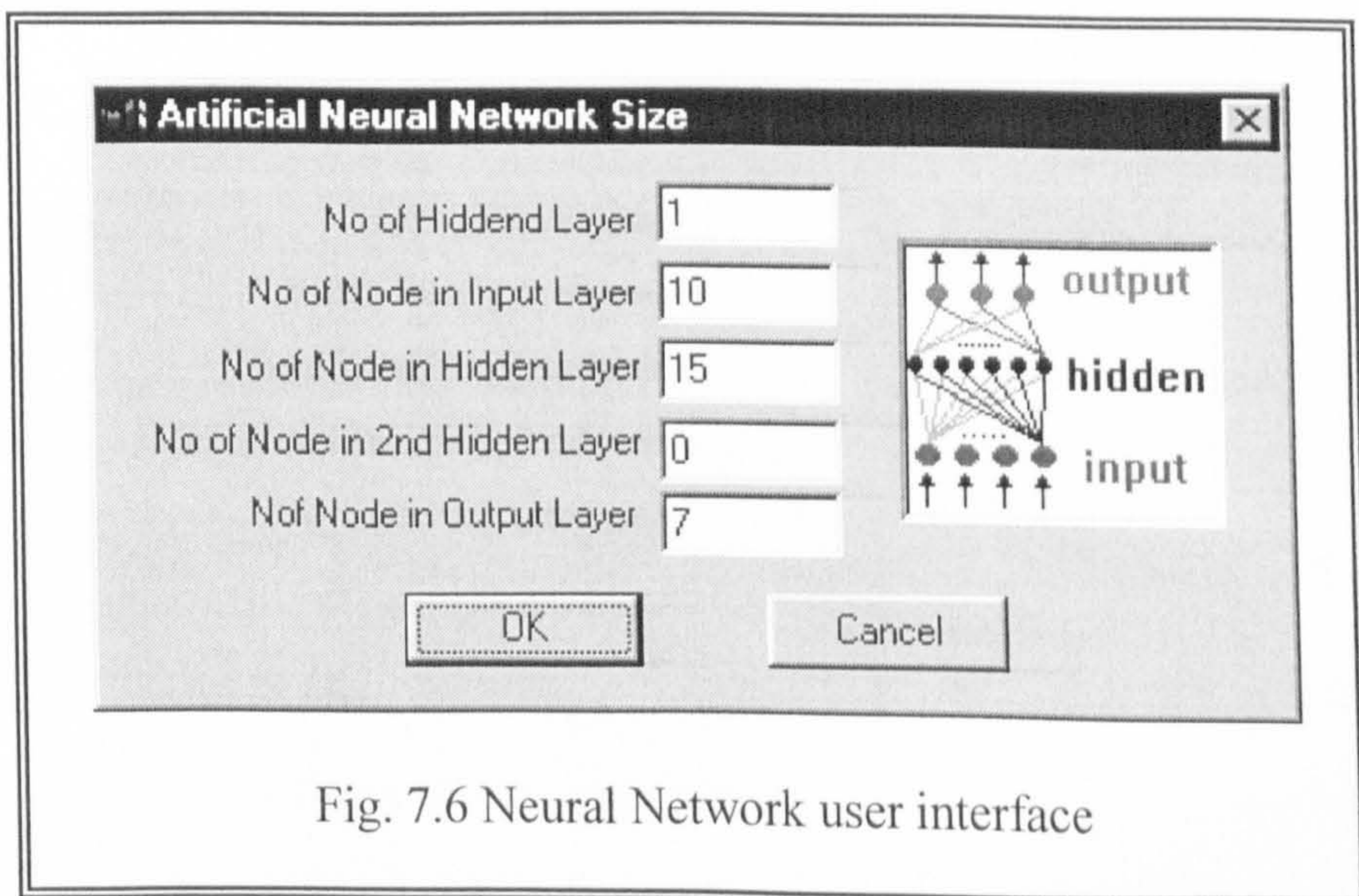


Fig. 7.6 Neural Network user interface

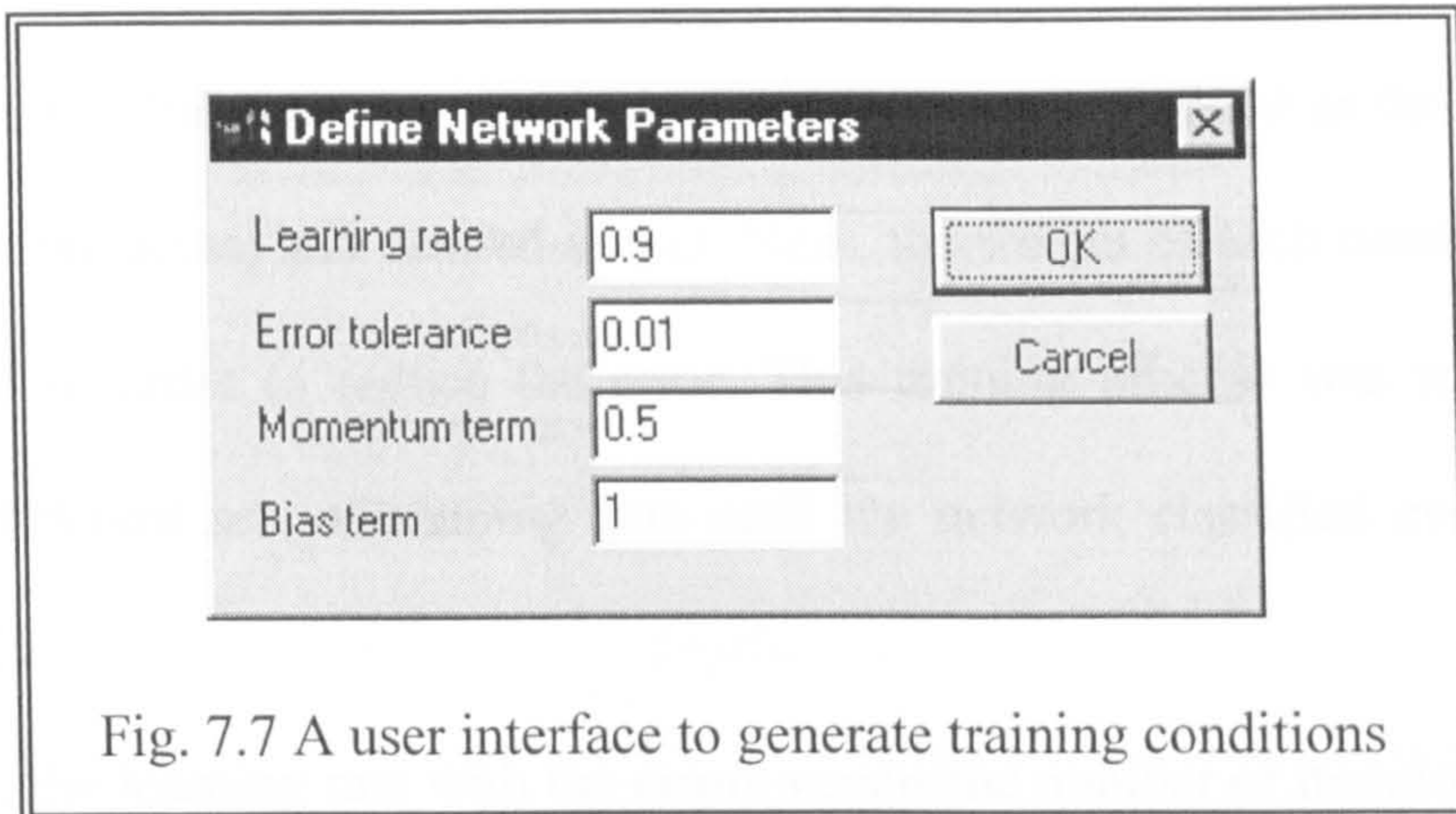
Each input layer neuron corresponds to a set of input geometric attributes. The bias unit is connected to all the neurons of the hidden layer and output layer. The number of neurons in hidden layer is generally based on prior experience or simply chosen arbitrarily and then refined by a result of trial-and-error. If there are not enough hidden neurons, the network is not able to learn the training parts; however, if too many hidden neurons are assigned to a network, each neuron adds to the overhead on the *CPU* during training and even the network loses the ability to generalise. In this research, the number of hidden neurons was arbitrarily determined as few as possible at first, and then it was added up until the network converge to a solution.

Networks with more hidden layers were also examined. The results were identical to the single hidden layer network, but the training times were considerably longer. The number of neurons in output layer corresponds to the number of feature in the feature library that this application deals with, such as *holes, pockets, bosses, blocks, slots and steps*.



## 7.4.2 Training

Figure 7.7 shows the dialogue box to define the training parameters, which are precisely discussed in Appendix 4.



In this research, the back-propagation learning involved with  $\eta$  (learning rate) = 0.9,  $\mu$  (momentum term) = 0.5,  $X_0$  (bias term) = 1.0, and  $\varepsilon$  (error tolerance) = 0.001. With these parameters and appropriate training patterns, the network was successfully trained.

The standard structure of ANNs is shown in Fig. 7.2. Each output feature is associated with a set of input geometric attributes. An array of 10 digit is used as the formation of input training attributes. The first seven input digits occupying from  $X_1$  to  $X_7$  is related to the chain code parameters. If the chain codes of the selected segment are '3 2 1 2 5 4 3', then first chain code '3' is assigned to  $X_1$  at the first neuron of input layer (*i.e.*,  $X_1 = 3$ ). In the same manner, all other chain codes are input to the following neurons,  $X_2 = 2, \dots, X_7 = 3$ . Other three input attributes such as circular/rectangular, convex/concave and



open/closed digit are presented to  $X_8$ ,  $X_9$ , and  $X_{10}$ , respectively. The detailed algorithms for deriving each attribute will be discussed in the section 7.5.

In this research, after presenting a full set of input training attributes, the actual activity of the six output neurons was individually compared with the desired feature type. The error was then calculated, which was defined as the difference between the actual and desired output. Next, the weight of each connection was changed in order to reduce the error. This training process was repeated for many different sets of training data until the network classified every feature correctly.

Finally, the learning rate with the errors versus the number of iteration is shown in Fig. 7.8.

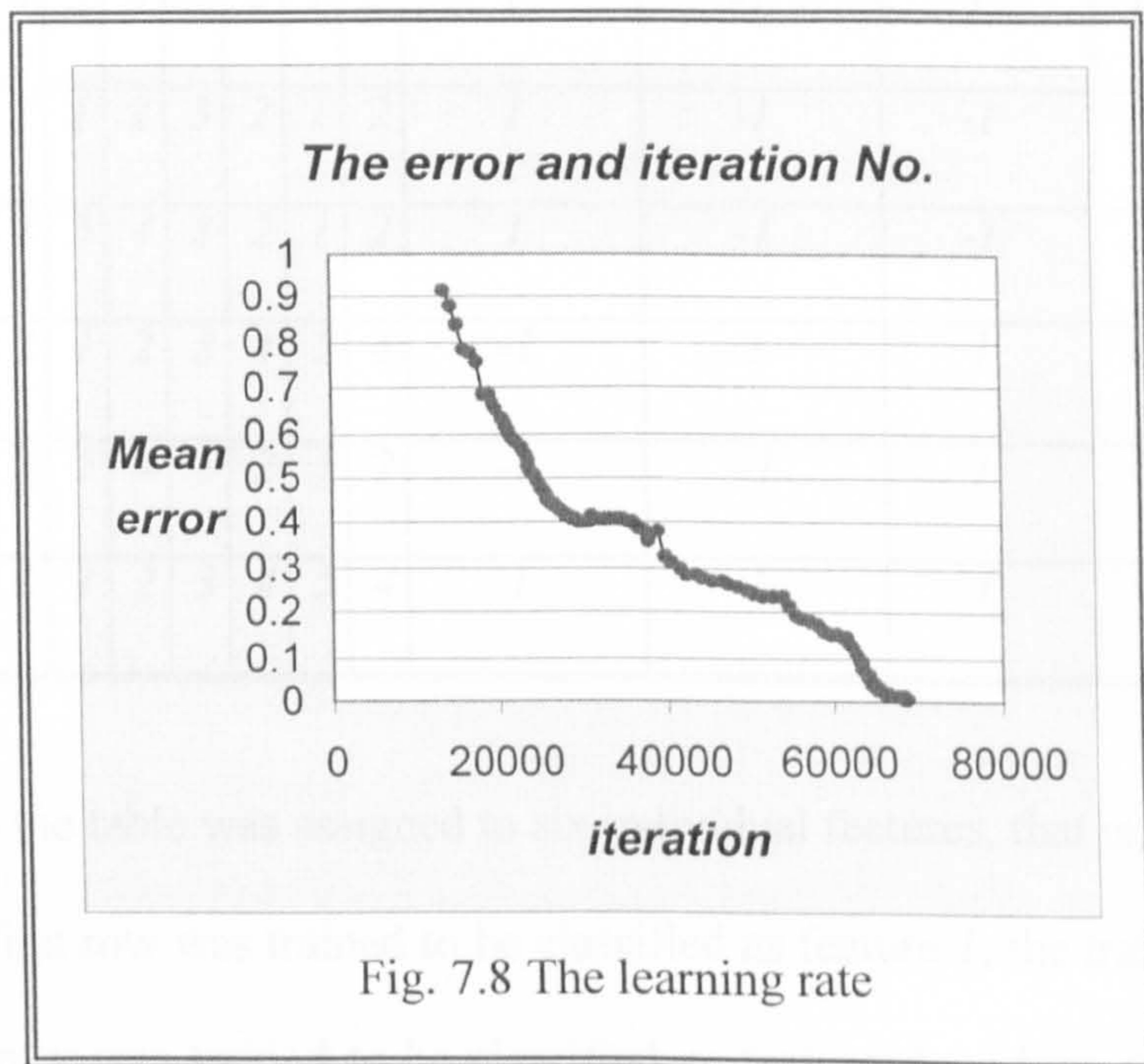


Fig. 7.8 The learning rate

It shows that the ANN module was trained completely after the all training data were presented to update the weights of connections with around 70,000 times

without any error. The total error typically drops quickly during the initial iterations but easily becomes destabilised. As the error converges toward zero, the rate of change in error gradually decreases.

Twenty-four training data were used in the training process using batch training which weight adjustments are made after all the training patterns are presented to the network in each iteration. Sampled training data used in this research are shown in Table 7.1.

Table 7.1 A training attribute (one epoch)

	<i>Chain Codes</i>							<i>circular</i>	<i>convex/</i>	<i>open/</i>	<i>Desired Output</i>
	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>/rect.</i>	<i>concave</i>	<i>closed</i>	
<i>Pocket</i>	4	5	4	3	2	1	2	1	-1	1	1,0,0,0,0,0
<i>Step</i>	2	1	2	3	2	1	2	1	-1	-1	0,1,0,0,0,0
<i>Slot</i>	4	5	4	3	2	1	2	1	-1	-1	0,0,1,0,0,0
<i>Boss</i>	2	1	2	3	4	5	4	-1	1	1	0,0,0,1,0,0
<i>Hole</i>	4	5	4	3	2	1	2	-1	-1	1	0,0,0,0,1,0
<i>Block</i>	2	1	2	3	4	5	4	1	1	1	0,0,0,0,0,1

Each row in the table was assigned to six individual features, that is, the training data of the first row was trained to be classified as feature 1, the training data of the second row was trained to be classified as feature 2, and so on. To train a pocket feature, input training attribute '4 5 4 3 2 1 2 -1 1' and desired output '1 0 0 0 0' is input to the ANN. This represents that the feature is a pocket, it has a set of chain codes (4 5 4 3 2 1 2), concave (-1), rectangular (1), and closed (1).



All the weight values in the network is updated until it produce valid output value in each output neurons (1 0 0 0 0 0). In the same fashion, all other training attributes are presented one by another.

After the network had been fully trained, all of the test features were successfully classified into their corresponding features on the basis of their geometry. That is to say, once users selected a portion of scanned point on the displayed image, the input geometric attributes of selected scanned point were automatically derived. They were then presented to the input layer of the ANN module and propagated through the various layers. ANN module analysed the attributes with its previous taught knowledge (weight values). It classified the input scanned points into an appropriate feature type according to the output value of the neuron that was high, whilst all the others were low. For instance, if the output value of the second neuron is the highest, then, the type of recognised feature must be a STEP as shown in table 7.1.

If more than one output is high, or none of the outputs is high, then the input attributes presented to the network does not belong to the taught feature classes. The choice is one of declaring a misclassification or simply assigning the attribute to the class of the output neuron with the highest numerical value (Gonzalez, 1992). ReTool system takes the highest numerical value.



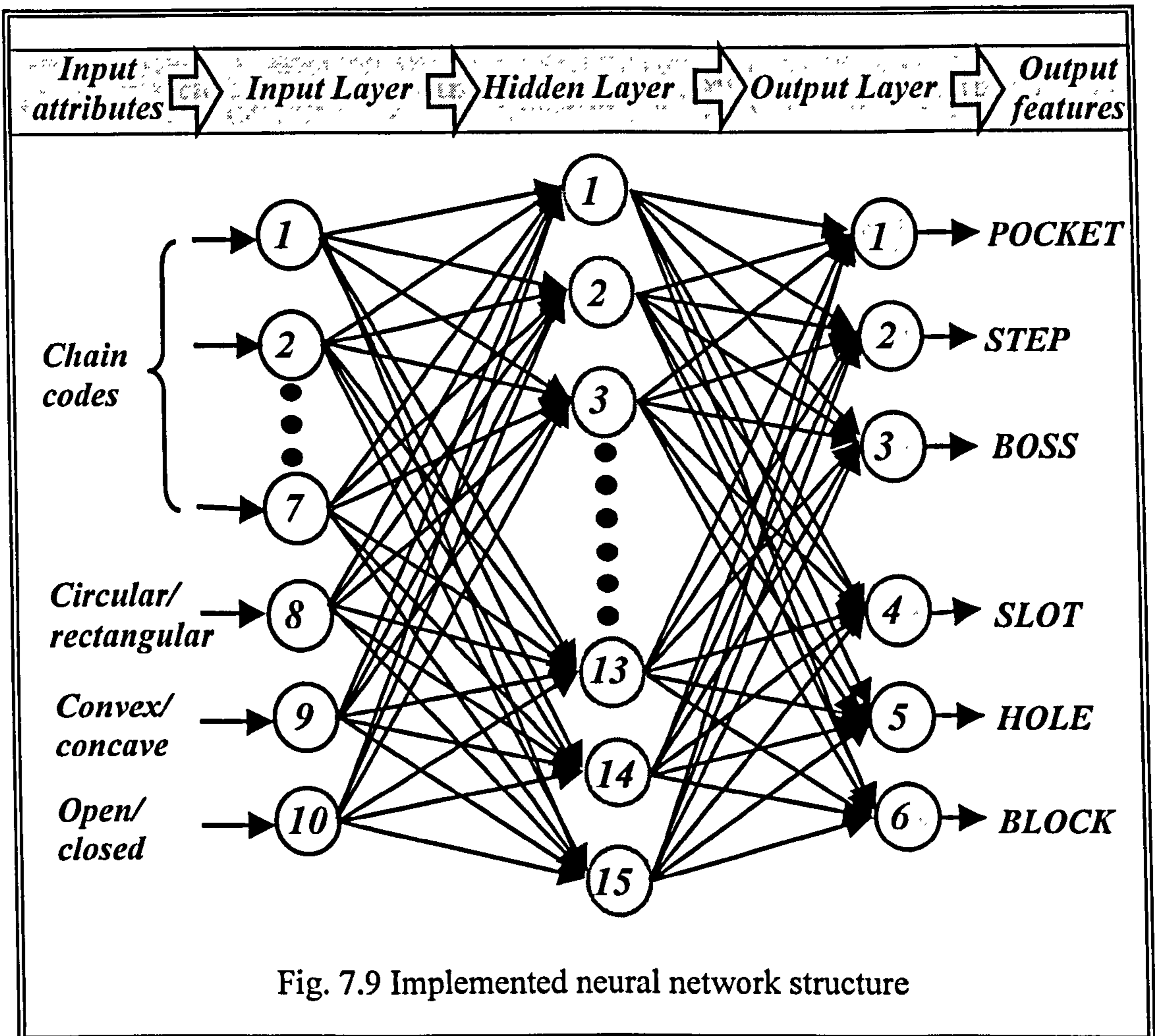
## 7.5 Deriving Geometric Attributes for Feature Recognition

A scanned point is a discrete entity having no topological relationship to any other neighbour points. Geometric attributes fetched from a point data set are vital for further feature recognition. That is, input attributes to the ANN from a point set are required to be chosen to contain as much information as possible, and that the information also need to be understood by the ANN. The premise of this research is that feature recognition is easier when performed on 2D attributes of 3D parts. These key input geometric attributes are extracted from the topology and geometry of a point set.

There are two basic properties of a successful representation scheme for input attributes (Nezis, 1997):

- It must be in a format that can be identified by the input layer of an ANN that can be implemented in computer software or hardware.
- There must be no overlap in the representation patterns. This means that the representation pattern of two different features must be different.

These attributes that help in feature identification are *chain codes*, *circular/rectangular*, *convex/concave* and *open/closed attributes*. They are the input values of the present ANN module, as shown in Fig. 7.9.



Each input attribute can be defined on arbitrary 2D plane. Hence, it can be seen that 3D geometric information can be produced by combining some of different 2D information. Recognising each feature requires the determination of these attributes. Once the four attributes are determined, a given segment may be uniquely recognised as a valid feature. New and reliable algorithms are developed for determining the attributes for each of the features. The algorithms are based on a set of general vector calculations from 3D scanned points. In the following subsections, the detailed algorithms for deriving each input attribute are discussed.

### 7.5.1 Chain code attributes

Chain codes approximately represent a section profile of scanned point data using a connected sequence of straight line by connecting every two points. Simple patterns can be concatenated to give compound patterns. This method is originated from the syntactic pattern recognition, which is a classical technique for recognising shapes from raster images (Gonzalez, 1992).

In this research, the number of chain code characters for representing the profile of a feature is fixed to *seven*. That is, a collection of seven chain codes needs to be derived corresponding with line segments of varying orientation as shown in Fig. 7.10 (a). It has a clockwise direction. For instance, Fig. 7.10 (b) shows chain codes that represent the section profile of a wedge feature. The number of chain code characters was simply chosen arbitrarily and then refined by a result of trial-and-error. That is, it was arbitrarily determined as few as possible at first, and then it was added up until the network can recognise all the target features.

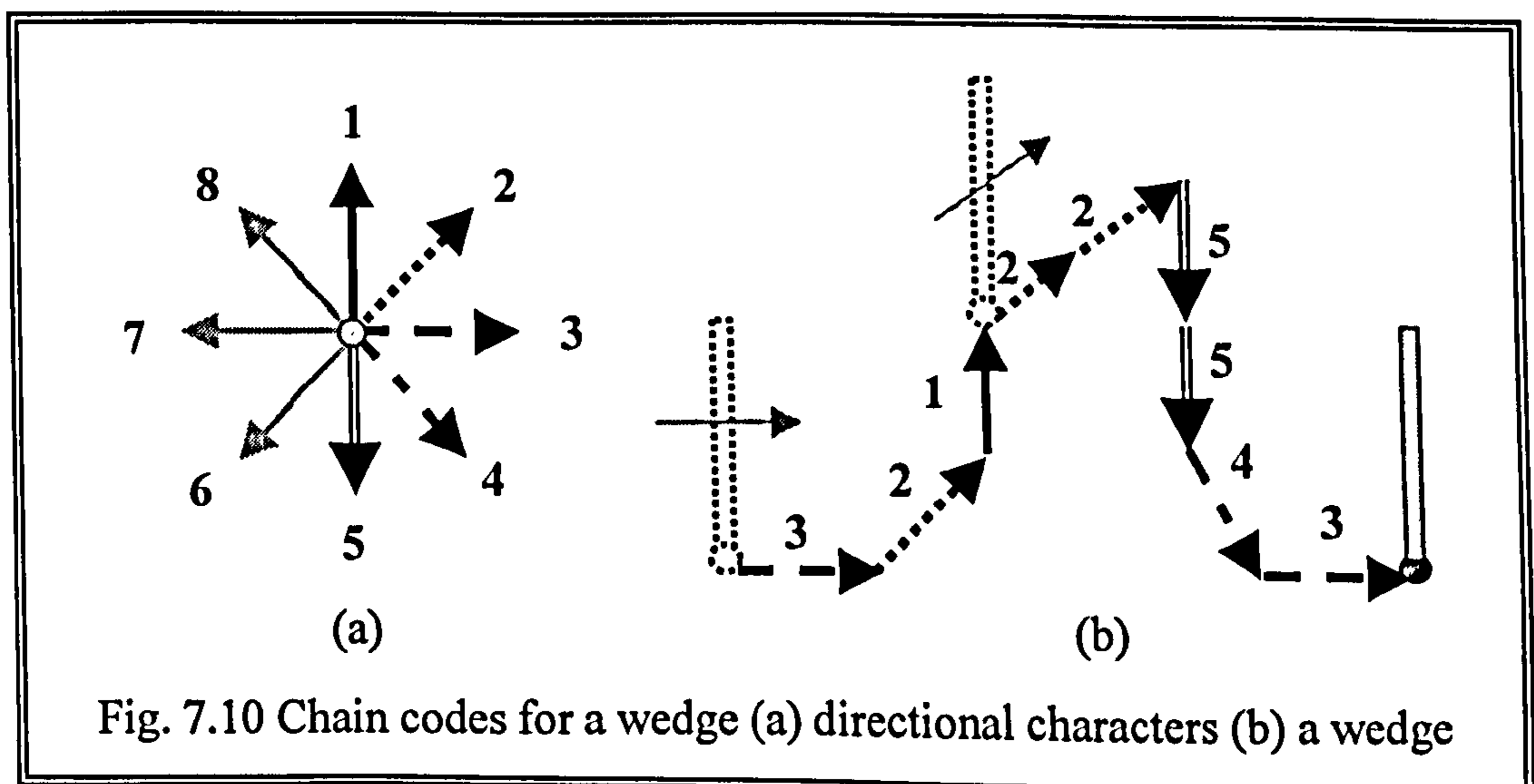
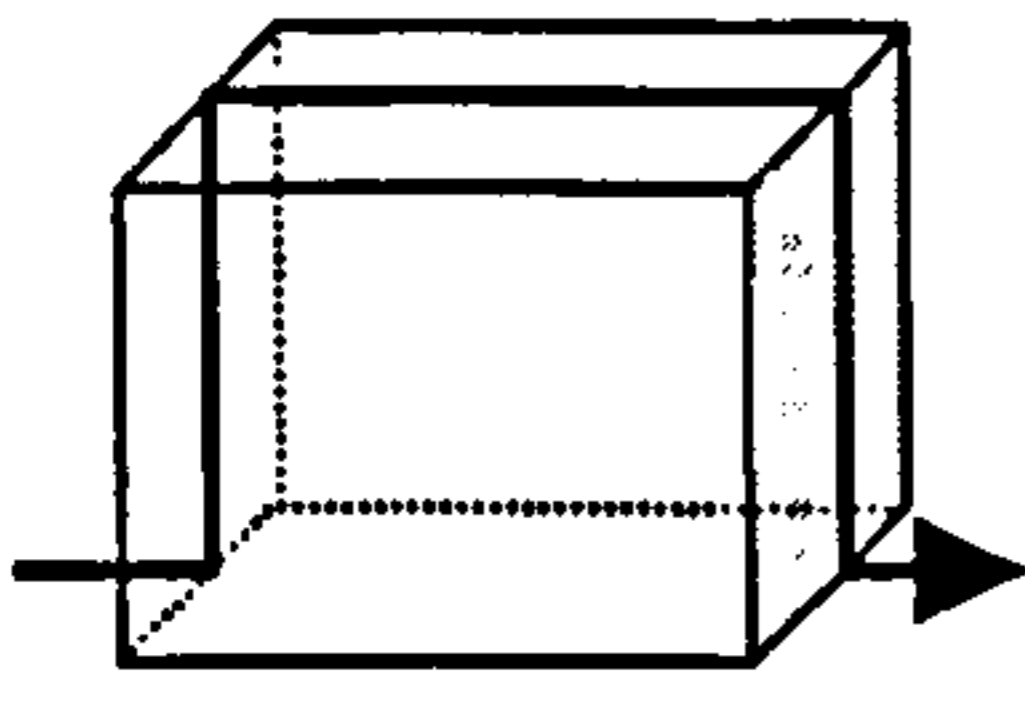
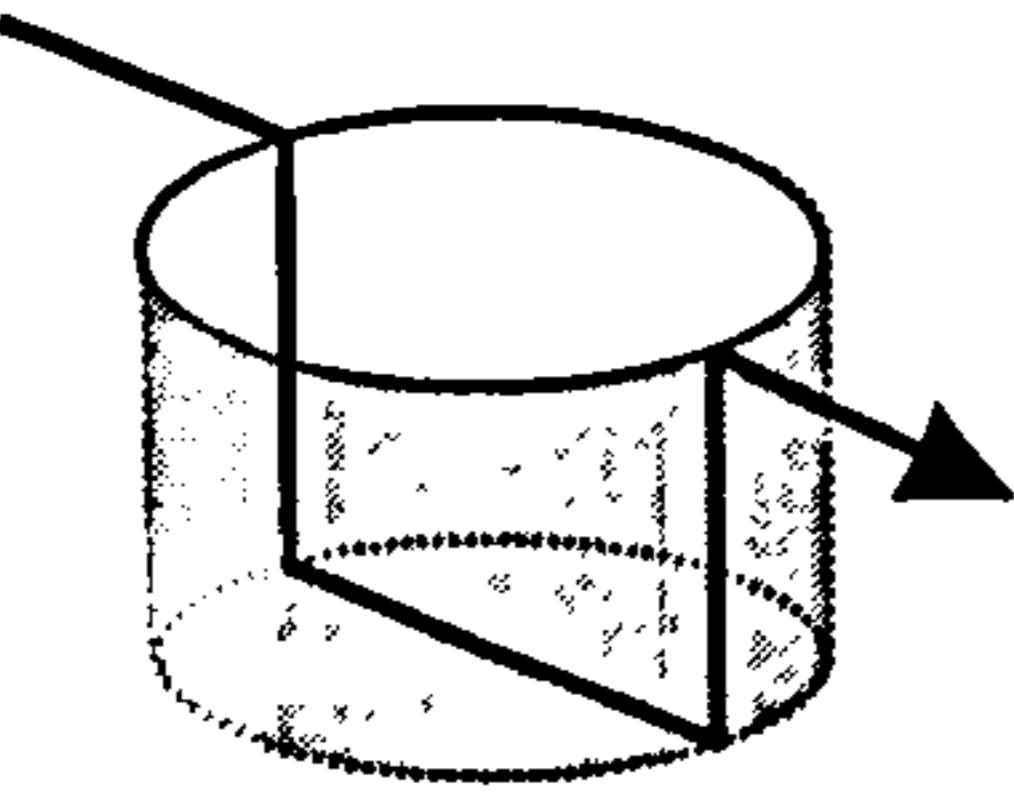
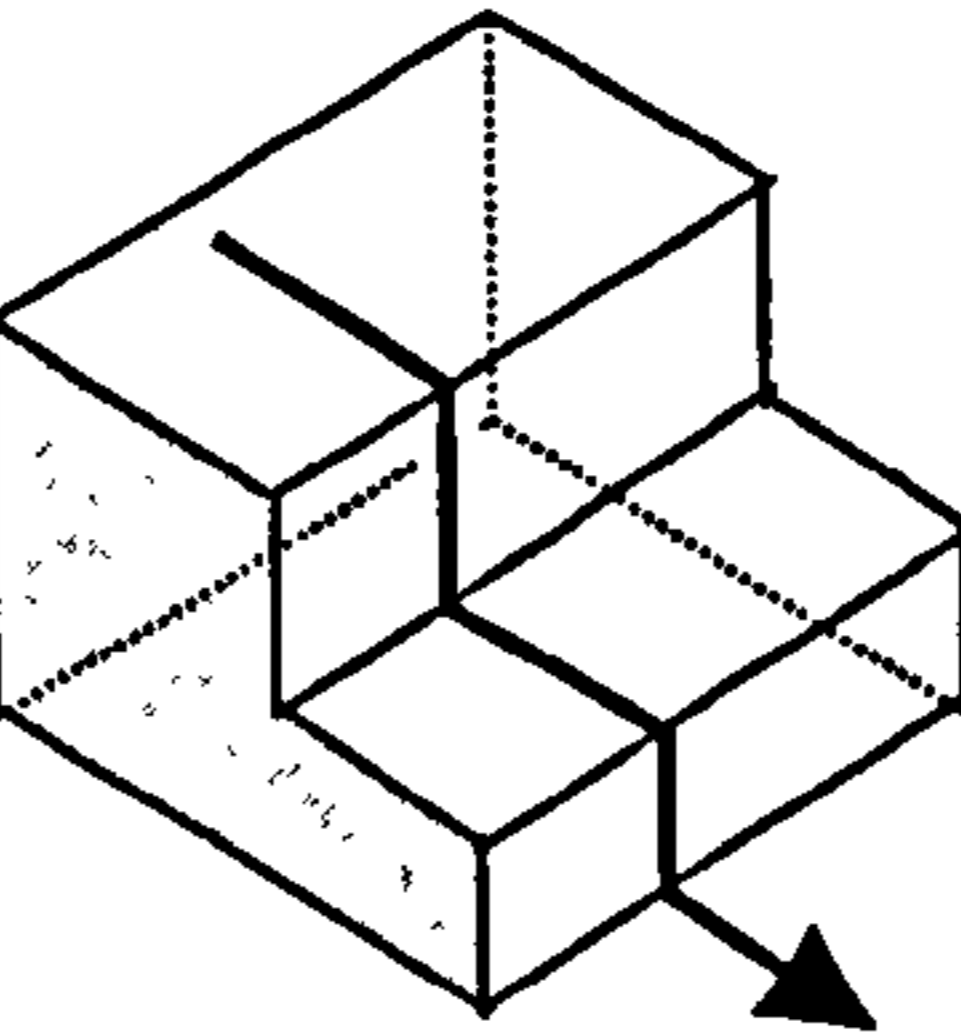
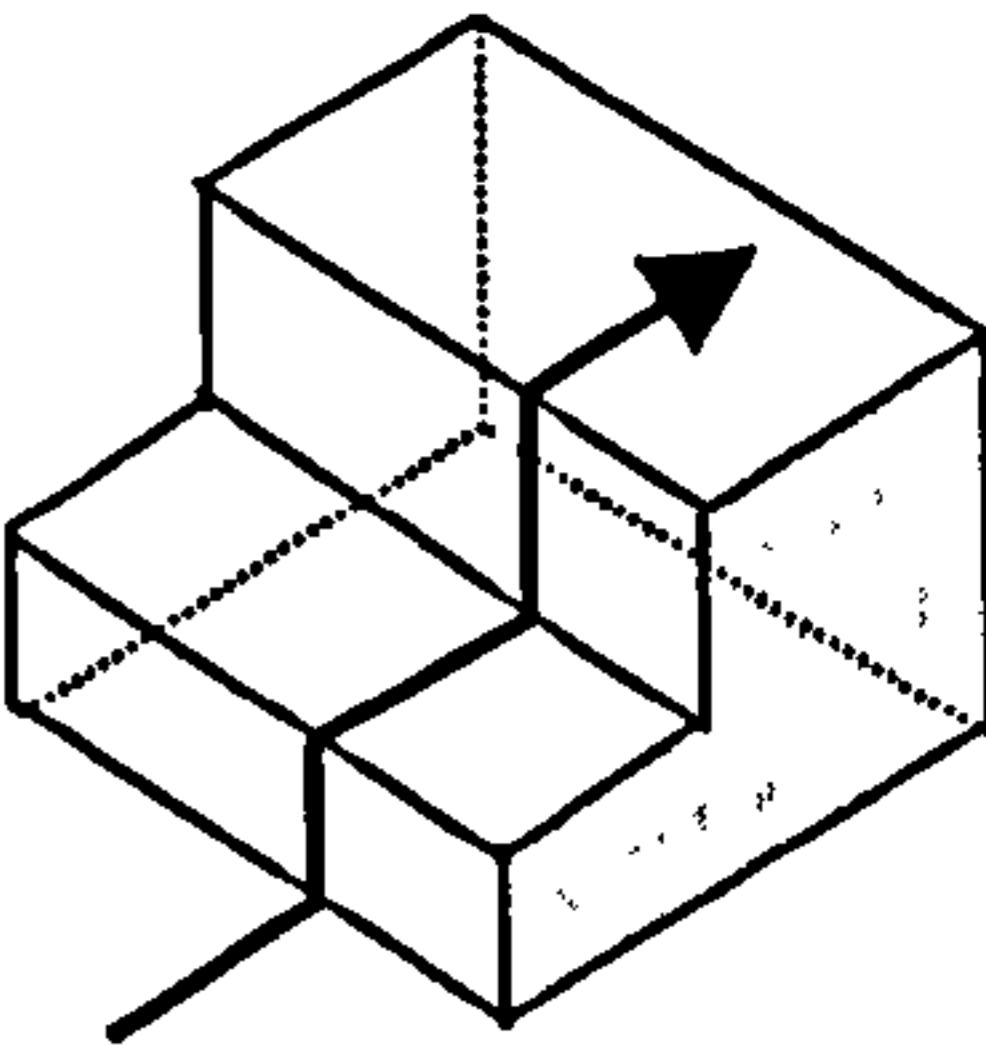
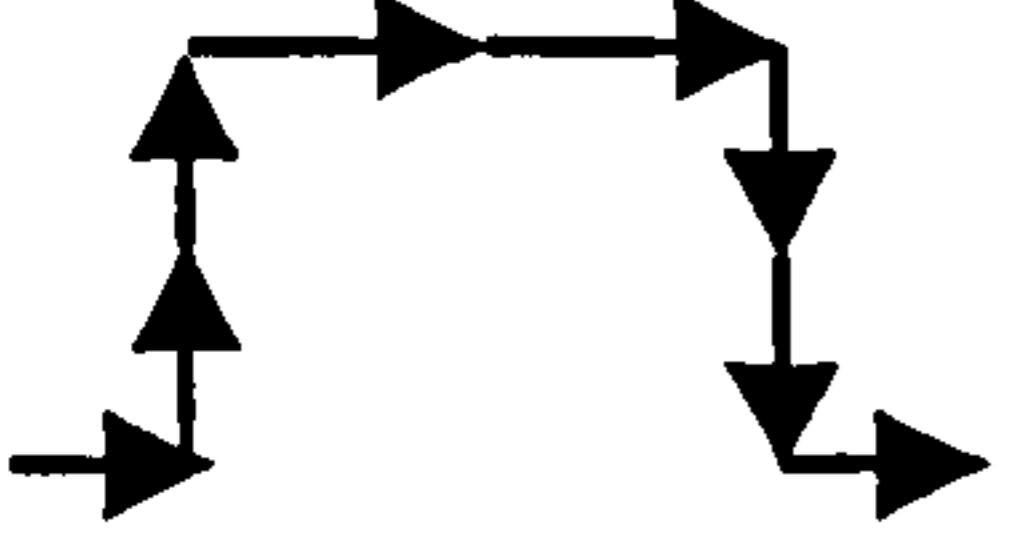
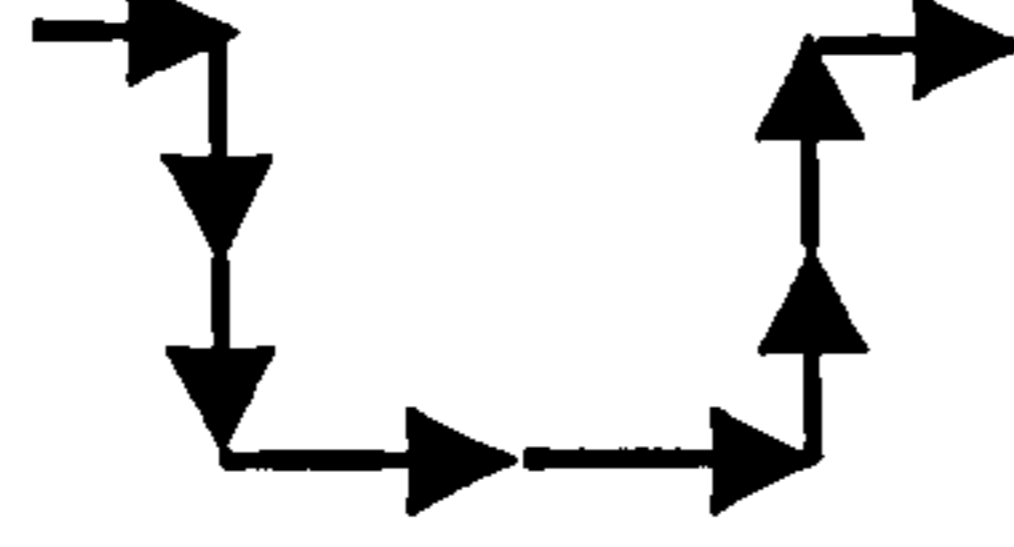
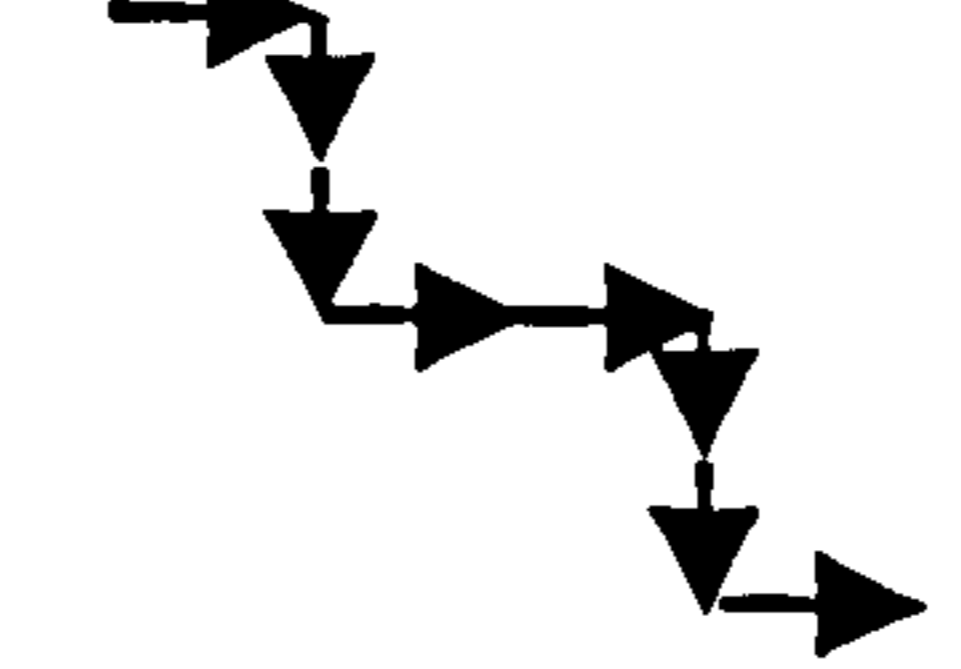
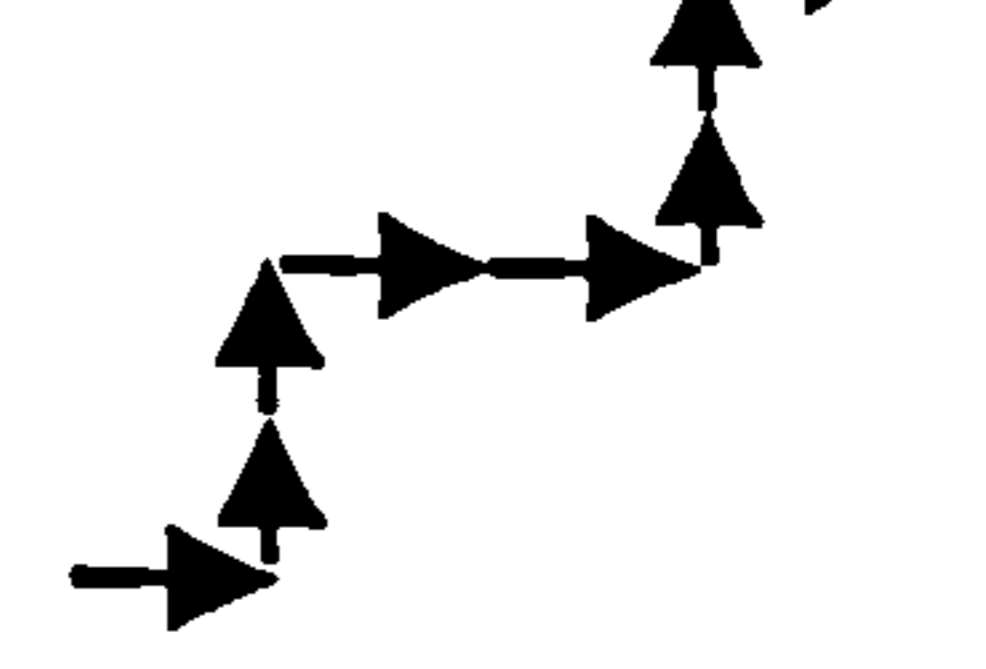
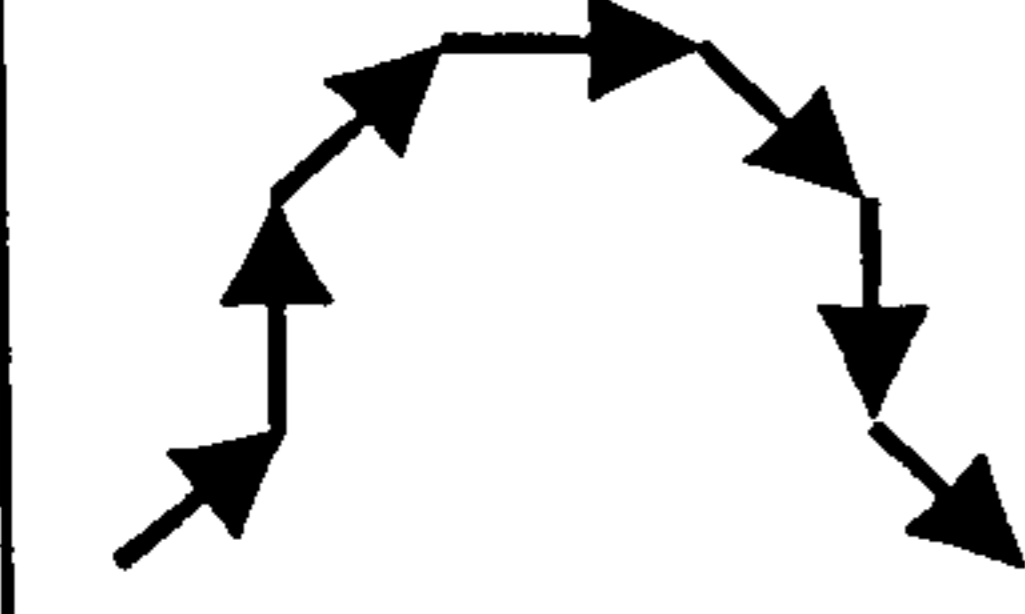
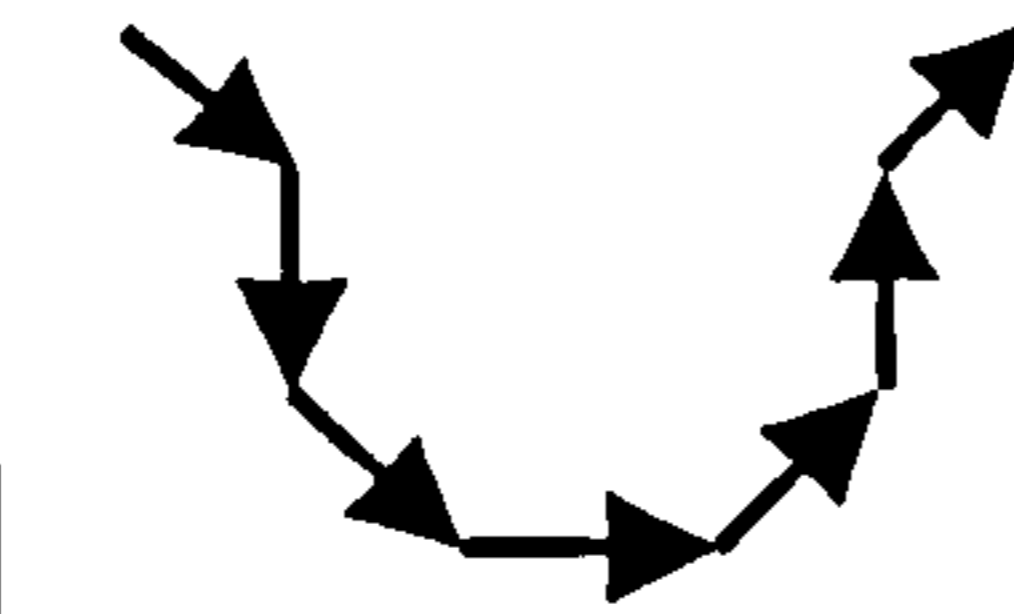
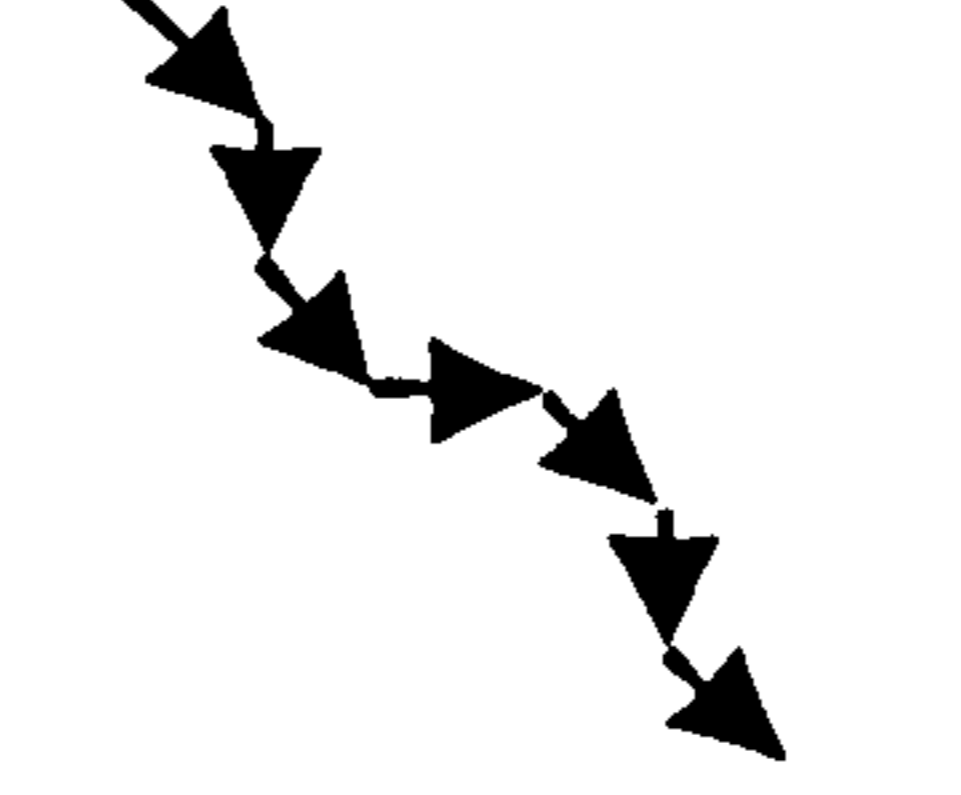
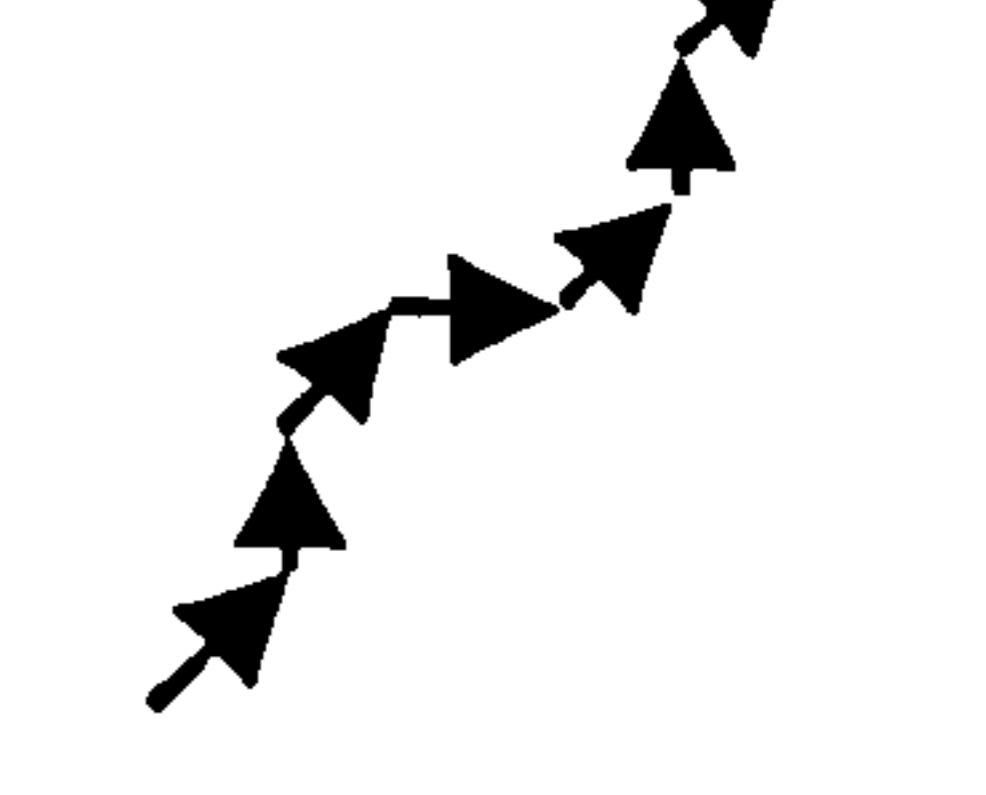


Fig. 7.10 Chain codes for a wedge (a) directional characters (b) a wedge



Symbolically, the chain codes of above example wedge are expressed by seven directional characters, as '3 2 1 2 5 4 3'. Each line segment in the section profile was replaced by the closest direction character using vector manipulations.

In the same manner, Fig. 7.11 shows the section profile of common generic features such as blocks, pockets, holes, bosses, steps, and slots.

		<i>Downward scan</i>	<i>Upward scan</i>
			
 <p><b>3 1 3 5 3</b></p>	 <p><b>3 5 3 1 3</b></p>	 <p><b>3 5 3 5 3</b></p>	 <p><b>3 1 3 1 3</b></p>
 <p><b>2 1 2 3 4 5 4</b></p>	 <p><b>4 5 4 3 2 1 2</b></p>	 <p><b>4 5 4 3 4 5 4</b></p>	 <p><b>2 1 2 3 2 1 2</b></p>
<i>Block / Boss</i>	<i>Hole / Pocket / Slot</i>	<i>Step</i>	
<p>Fig. 7.11 Example chain codes for target generic features</p>			

Using a CMM, the scanning direction is often set to be parallel to one of vertical planes in the Cartesian coordinate system. Accordingly, the direction of chain codes is also parallel to the vertical plane of the scanning direction.

Let us examine how to derive chain codes from a set of scanned points. Suppose a given set of scanned data points  $R_i$ , where  $i = 1, \dots, n$  ( $n$ : the number of points in the selected scan line), as shown in Fig. 7.12.

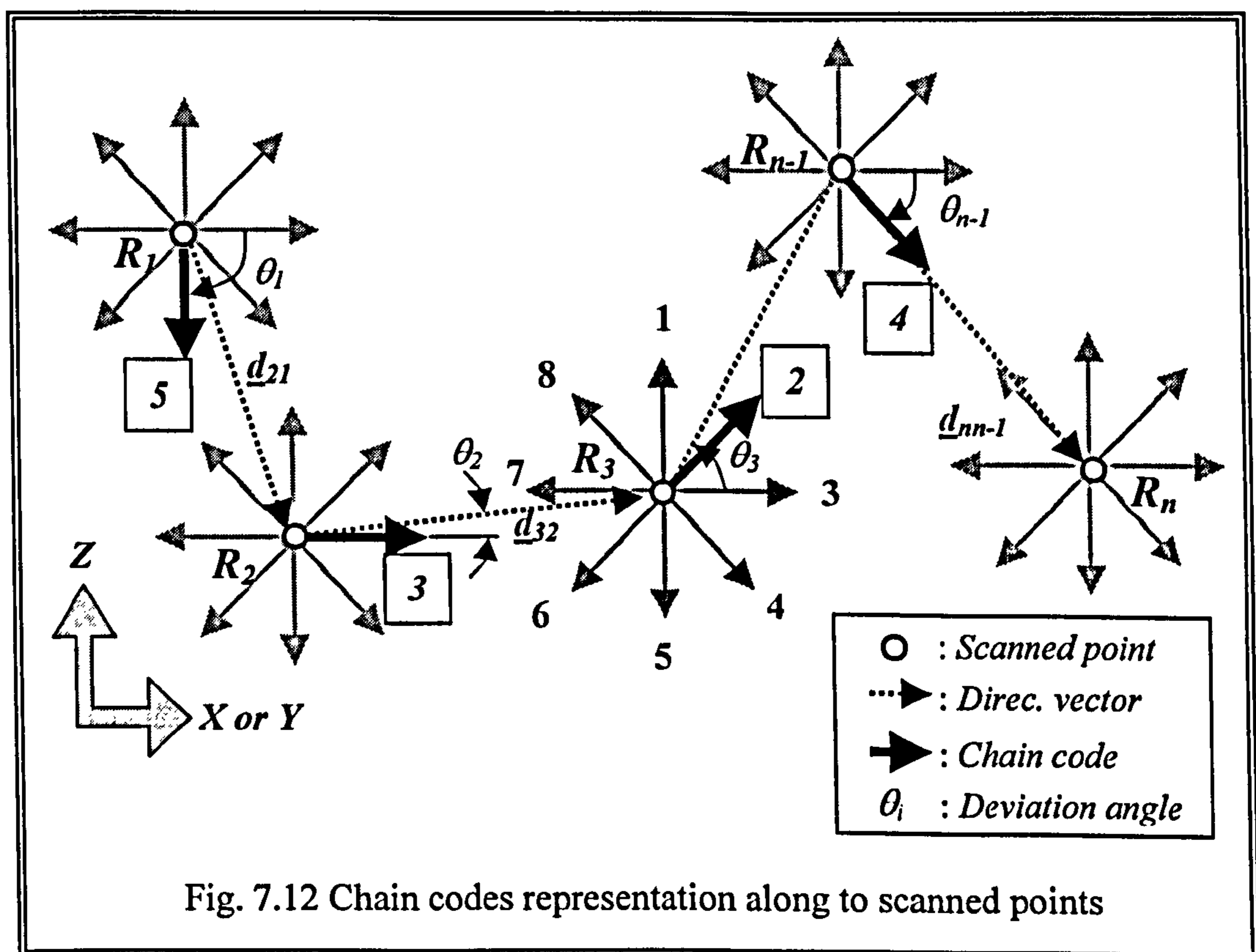


Fig. 7.12 Chain codes representation along to scanned points

From the given set of scanned data points, the first ( $R_1$ ) and the second ( $R_2$ ) point data are assigned to the developed module to compute their directional vector,  $d_{21}$ , connecting  $R_1$  and  $R_2$ . Then, the deviation angle  $\theta_1$  between computed directional vector ( $d_{21}$ ) and one of horizontal axis vectors ( $x$  or  $y$ -axes) is calculated. By examining the angle  $\theta_1$ , the most parallel chain code to the

considered directional vector can be determined. For instance, the direction of vector  $\underline{d}_{21}$  at  $R_1$  is the most parallel to that of '5' chain character, then the chain code between  $R_1$  and  $R_2$  is registered as '5'. All other scanned points,  $R_2, R_3, \dots, R_n$ , are evaluated by the same process until a full set of chain codes are obtained.

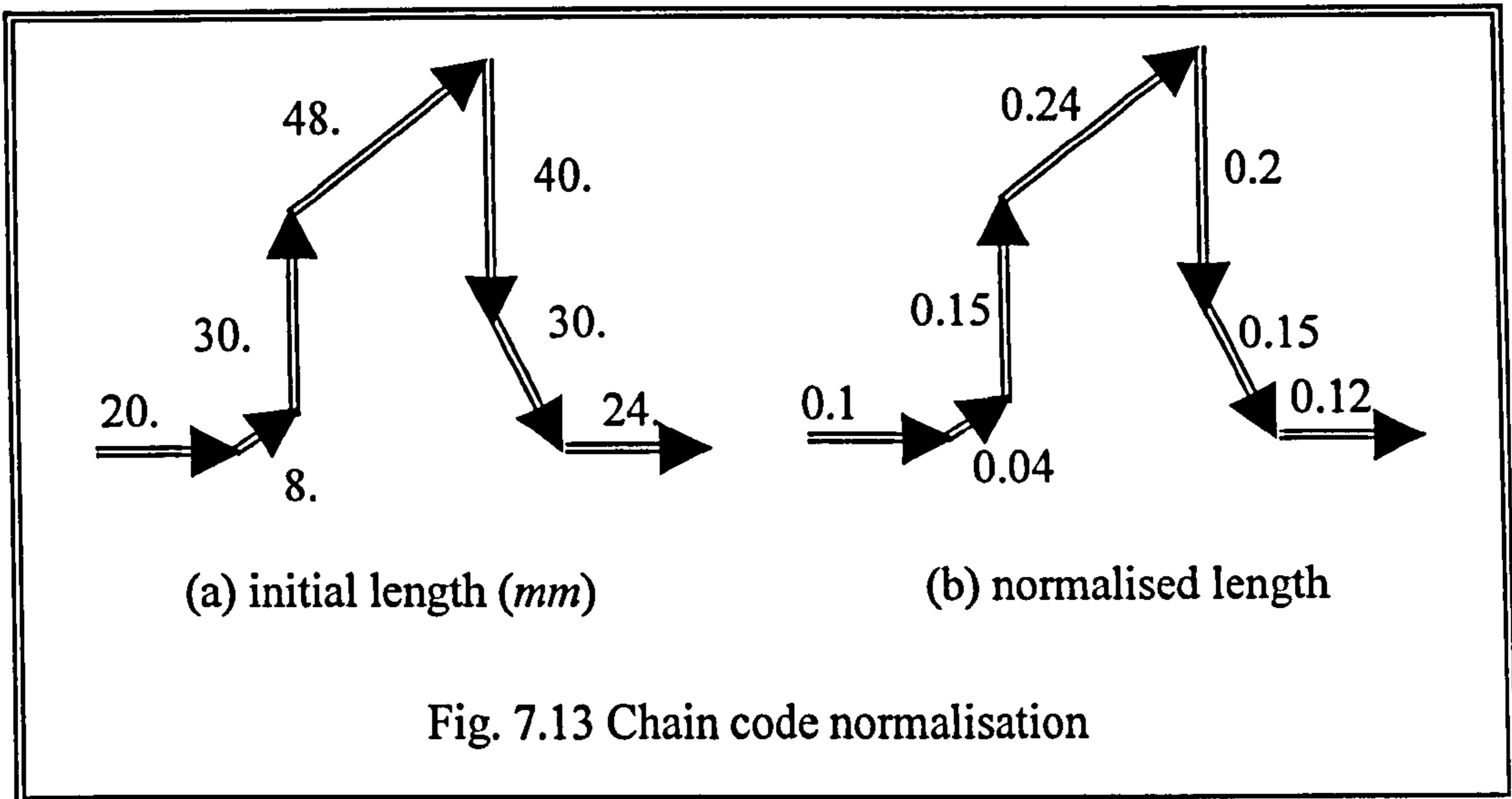
The accuracy of the resulting chain codes surely depends on the complexity of parts and on the accuracy of the scanners. Therefore, to implement the chain code techniques to the application, the following two major drawbacks needed to be overcome at first (Gonzalez, 1992).

- The resulting chain codes is often long
- Making a *chain code* is noise-sensitive

In this research, to overcome the shortfalls, each scan line was normalised by setting the maximum length whilst proportionately scaling all segments between every pair of chain codes. This normalisation technique gives a crucial solution toward *size-invariant* feature recognition. If all consecutive chain codes that have same direction are concatenated, the number of chain codes for simple features can be less than seven (direction characters) as mentioned in the beginning of this section.

In addition, some of error points can be neglected with this normalisation technique. This benefit is acquired by tagging any point which normalised length is less than certain criterion. This is based on the fact that error points are likely to have short normalised length. Figure 7.13 shows the normalisation results of a wedge feature.





### 7.5.2 Convex/concave attribute

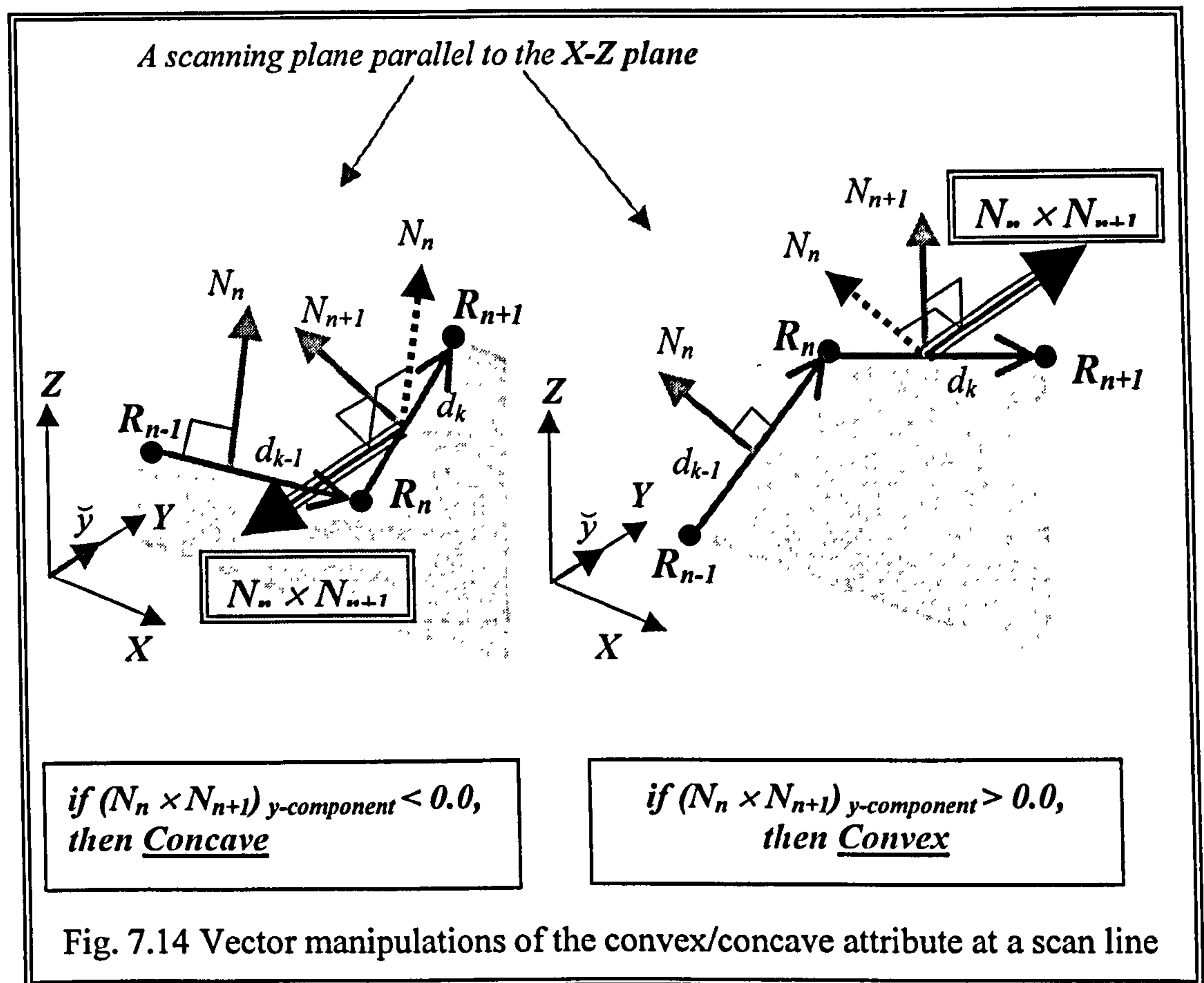
This attribute represents whether a feature is considered convex or concave. The nature of this information is internally contained in the chain code attribute. However, if this information can be directly handled, it is even more straightforward to train the ANN. This attribute is determined at the same time when chain codes are derived, because it also deals with same edge points on the same vertical plane.

Convex is termed positive and concave is termed negative. Mathematically, a convex shape is that for which the angle between two entities is greater than  $180^\circ$ , whilst the angle of two entities of a concave shape is less than  $180^\circ$ .

Accordingly, a mathematical algorithm to derive this attribute is given as

follows. As shown in Fig.7.14, suppose that  $\sum_{k=1}^n R_k$  and  $\sum_{k=1}^{n-1} N_k$  denote a set of scanned points and their outward normal vectors lying on the scanned plane,

respectively. Assume the scanning path is parallel to the  $X$ - $Z$  plane in this example.



Using the line-segments connecting two consecutive scanned points, the  $k^{\text{th}}$  outward normal vectors,  $N_k$ , can be calculated: that is,

$$N_k = d_k \times \tilde{y} \quad (k=1, 2, \dots, n) \quad (7.3)$$

where  $d_k$  is the directional vector of a scan line and  $\tilde{y}$  is  $y$ -directional unit vector. The convex/concave attribute is determined by investigating the sign of the  $y$ -component of the vector resulted by the cross product of two consecutive normal vectors along to the strip line.

$$Sy_k \text{ (the sign of y-component)} = (\underline{N}_k \times \underline{N}_{k+1})_{y\text{-component}} \quad (7.4)$$

$\underline{N}_k$  and  $\underline{N}_{k+1}$  are  $k^{\text{th}}$  and  $(k+1)^{\text{th}}$  line segment. If there is any pair of two consecutive normal vectors which y-component of the *cross product* is less than 0.0 ( $Sy_k < 0.0$ ), the section profile of the feature must be “*concave*”. Otherwise ( $Sy_k \geq 0.0$ ), it is “*convex*”.

Figure 7.15 shows the procedure for computing convex/concave attribute. This criterion needs to be changed according to the scanning direction. For instance, if scanning direction is parallel to the Y-Z plane, the sign of x-component needs to be considered.



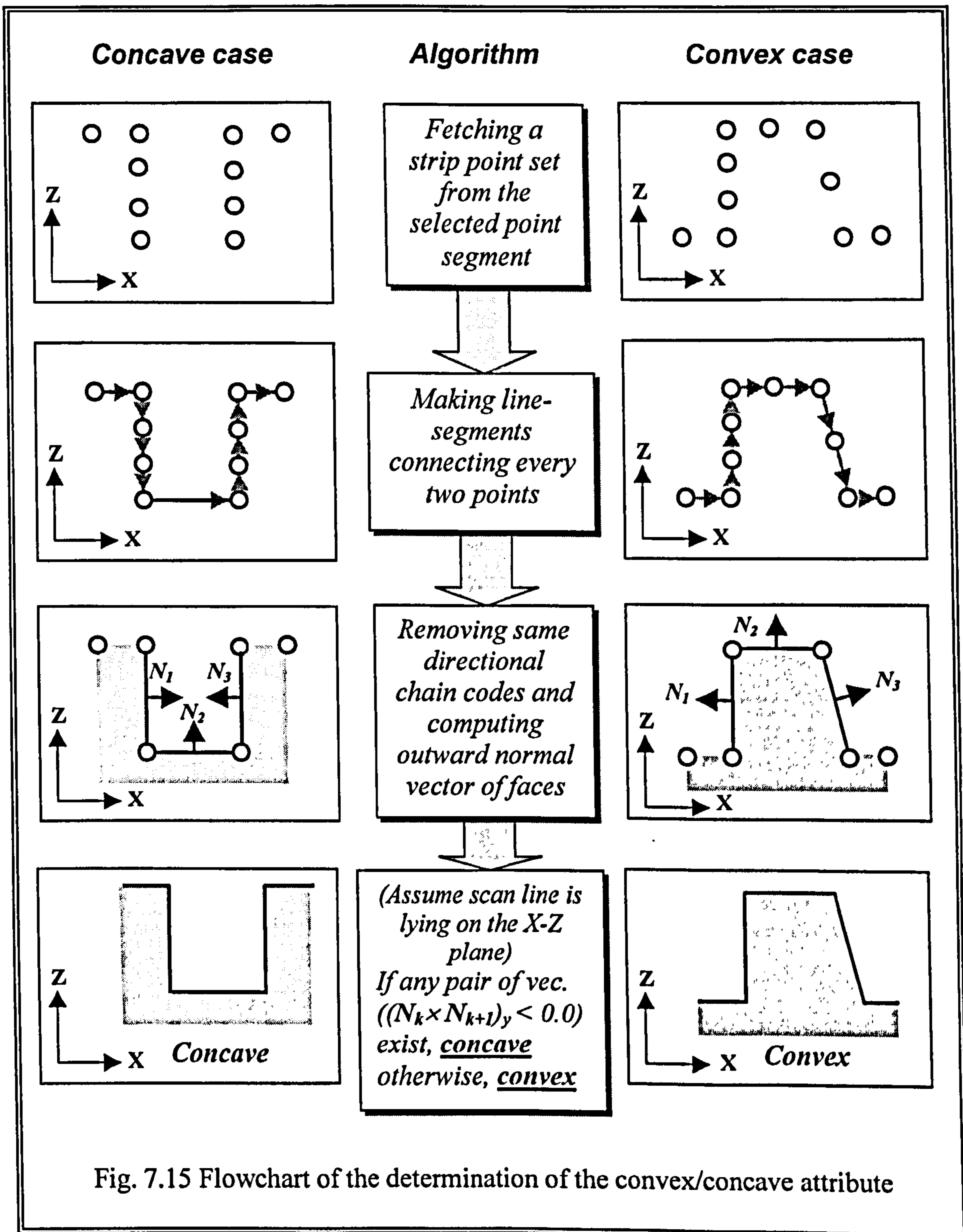


Fig. 7.15 Flowchart of the determination of the convex/concave attribute

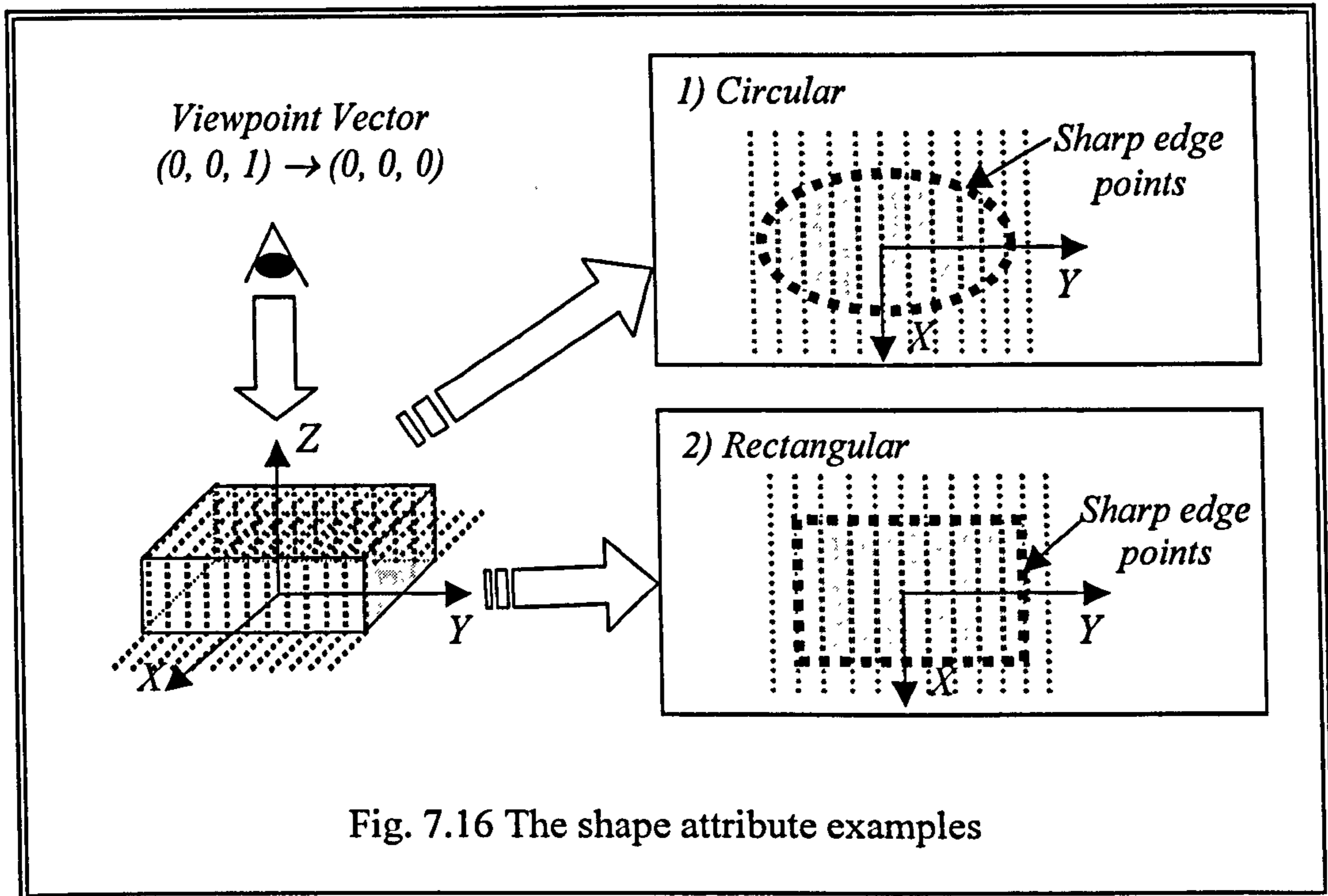
As shown in Table 7.2, a flag is set to '1' to identify a negative shape (a convex) or '-1' to identify a positive shape. It shows the possible attribute values according to the convexity. Therefore, if the selected points have a convex shape, the input neuron of the ANN is assigned by '1'. Otherwise, it is input by '-1'.

Table 7.2 Convex/concave attribute

<b>Features</b>	<b>Convex/concave pattern</b>	<b>Input Value to the ANN</b>
<i>Block and Boss</i>	<i>Convex</i>	<i>1</i>
<i>Step, Slot, Pocket and Hole</i>	<i>Concave</i>	<i>-1</i>

### 7.5.3 Circular/rectangular attribute

This attribute is used to determine whether the 2D geometric loop pattern of a feature on the  $X$ - $Y$  domain (viewing the part from top view) is considered *circular* or *rectangular*, as shown in Fig. 7.16.



In order to determine this attribute, the rim points of a selected feature are fitted into a circle at first. That is, the location of the centre of a circular arc and its radius are estimated. Then the maximum fitting error derived is used to determine if the unknown shape is circular or rectangular by comparing the error with a predefined error tolerance. If the calculated maximum fitting error is less than a predefined tolerance, the shape of a selected point set is circular. Otherwise, it is rectangular.

In this research, the Landau's methodology (1987) was used to fit the edge points to a circle. His method is based upon an algorithm that minimises the



radius and the centre estimation error of the circle based on the mean square error of selected point data. In this work, the method is extended to the principle of estimation 3D-circle centre and its radius.

The brief algorithm of Landau's methodology for estimating circular centre and radius is as follows. Given a set of point data  $(x_1, y_1, z_1), \dots, (x_N, y_N, z_N)$ , which represent the boundary rim of an unknown feature, the goal is to estimate the best set of given points using the LMS technique. These estimates are used for further iteration process. As shown in Fig 7.17,  $\underline{r}_1, \dots, \underline{r}_N$  are defined as the radius vector to each point for given a set of scanned point data  $(x_1, y_1, z_1), \dots, (x_N, y_N, z_N)$ .

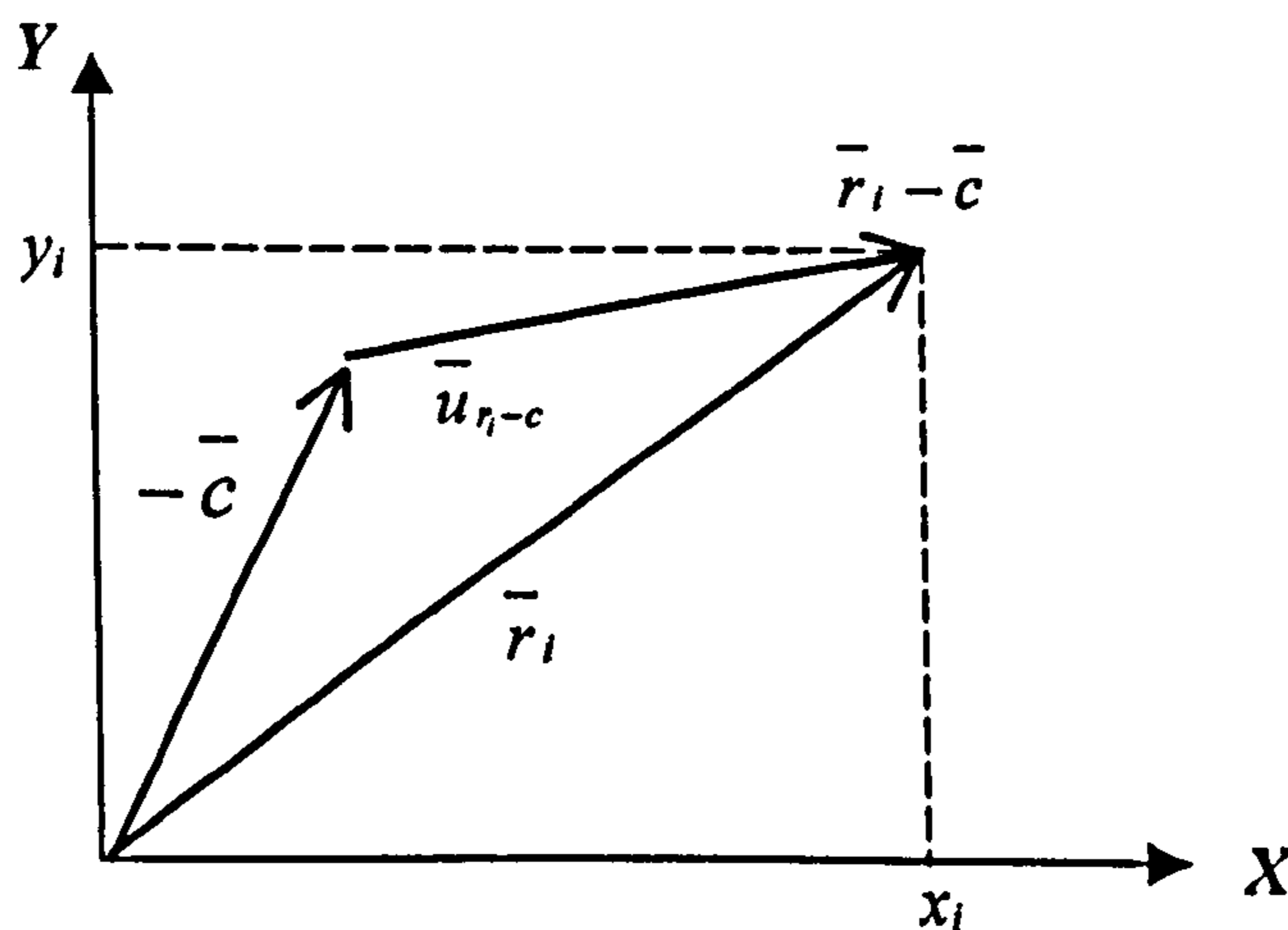


Fig. 7.17 A set of scanned points  $(x_i, y_i, z_i)$ , its arc centre and radius

In order to define a circular arc with a radius  $R$  and a centre at  $\underline{c} (x_a, y_a, z_a)$ ; the error  $\Delta r_i$  is defined as follow.

$$\overline{\Delta r}_i = \overline{r}_i - \overline{c} - R\overline{u}_{r_i-c}, \quad (7.5)$$

where  $\underline{u}_{r_i-c}$  is a unit vector in the direction of  $\underline{r}_i - \underline{c}$ .

Summing up the square of the errors, we have

$$Error(R, \overline{c}) = \frac{1}{N} \sum_i (\Delta r_i)^2 = \frac{1}{N} \sum_i (\overline{r}_i - \overline{c} - R\overline{u}_{r_i-c})^2 \quad (7.6)$$

$N$  is the number of scanned points associated with the corresponding circle. The error function  $Error(R, \overline{c})$  is the mean square error that must be minimised with respect to  $R$  and  $\overline{c}$ .

Therefore, differentiating (7.6) with respect to  $R$  yields

$$\frac{\partial}{\partial R} Error(R, \overline{c}) = \frac{2}{N} \sum_i (\overline{r}_i - \overline{c} - R\overline{u}_{r_i-c}) \bullet \overline{u}_{r_i-c} = 0 \quad (7.7)$$

Hence, the estimated radius

$$\begin{aligned} \sum_i \{|\overline{r}_i - \overline{c}| - R\} &= 0 \\ R &= \frac{1}{N} \sum_i |\overline{r}_i - \overline{c}| \end{aligned} \quad (7.8)$$

Differentiating (7.6) with respect to  $x_a$ ,  $y_a$ , and  $z_a$  yields,

$$\frac{\partial}{\partial x_a} Error(R, \overline{c}) = \frac{1}{N} \sum_i (\overline{r}_i - \overline{c} - R\overline{u}_{r_i-c}) \bullet u_{x_a} = 0,$$

$$\frac{\partial}{\partial y_a} \text{Error}(R, \bar{c}) = \frac{1}{N} \sum_i (\bar{r}_i - \bar{c} - R\bar{u}_{r_i-c}) \bullet u_{y_a} = 0,$$

$$\frac{\partial}{\partial z_a} \text{Error}(R, \bar{c}) = \frac{1}{N} \sum_i (\bar{r}_i - \bar{c} - R\bar{u}_{r_i-c}) \bullet u_{z_a} = 0$$

Thus, the vector  $\sum_i (\bar{r}_i - \bar{c} - R\bar{u}_{r_i-c})$  must be zero. Therefore, the estimated centre coordinates are as follows,

$$\bar{c} = \frac{1}{N} \sum_i (\bar{r}_i - R\bar{u}_{r_i-c}) \quad (7.9)$$

To solve equation (7.8) and (7.9) simultaneously, the following procedure was proposed in (Landau, 1987).

1.  $\underline{c}_k$  initialisation ( $k = 0$ ), where  $\underline{c}_k$  is the  $k^{\text{th}}$  estimated centre coordinate
2. Substitute  $\underline{c}_k$  in (7.8) and find  $R_k$ , where  $R_k$  is the  $k^{\text{th}}$  estimated radius
3. Substitute  $\underline{c}_k$  and  $R_k$  in (7.9) and find  $\underline{c}_{k+1}$
4. if  $|\underline{c}_k - \underline{c}_{k+1}|$  is less than predefined tolerance, then exit, else go to step 2

This algorithm converges to the true circle reasonably quickly with the best estimated circle centre and its radius. The convergence rate with the errors versus the number of iteration for an estimated circle is shown in Fig. 7.18.



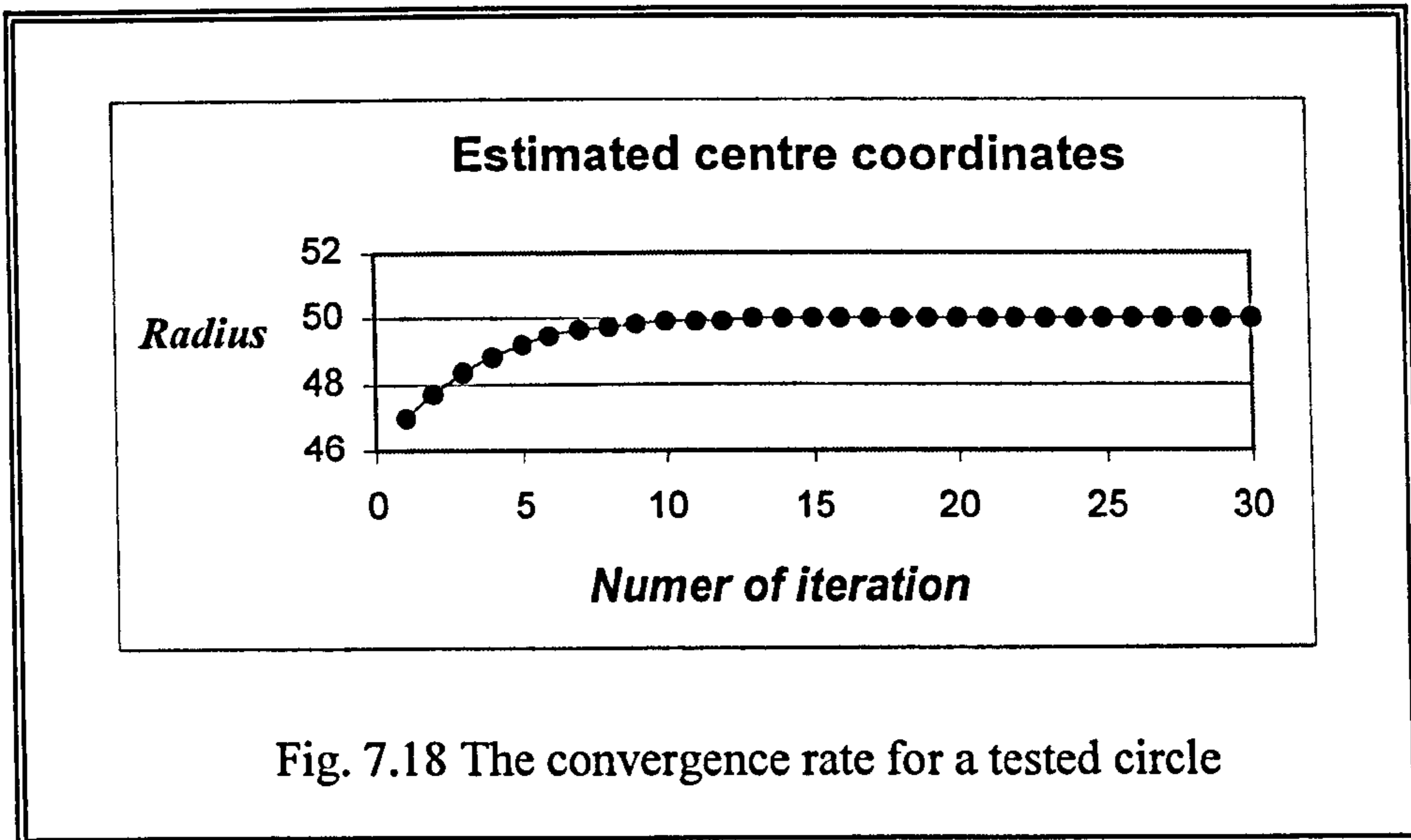


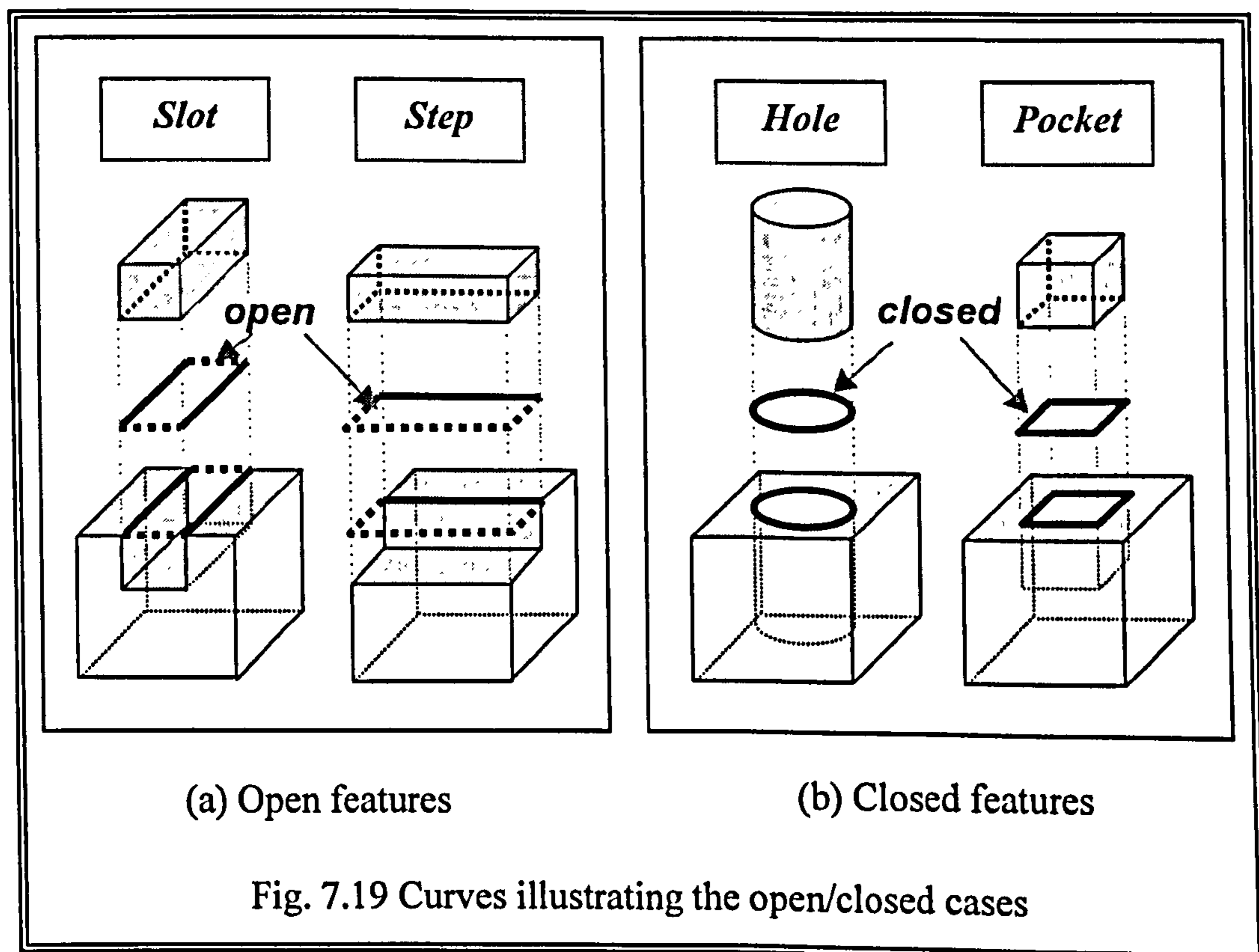
Fig. 7.18 The convergence rate for a tested circle

In order to determine this attribute, the maximum fitting error between the estimated radius and real distance from estimated centre point to all edge points needs to be calculated. If the calculated maximum fitting error is less than a predefined tolerance, the shape of a selected point set is likely to be circular. Otherwise, it is rectangular.

Currently, just two classifications are used in this work such as *rectangular (1)* and *circular type (-1)* depending on the number of detected edge points on the *X-Y* plane. However, this classification can be further changed or expanded for system versatility.

### 7.5.4 Open/closed attribute

The final attribute is the *open/closed attribute*, which is used to determine if a feature is considered closed or open. This is directly related to the number of imaginary faces present in the part. A feature is *closed* if the lines (edges) representing the boundary perimeter of its feature are all real and can form a continuous loop. If any edge is missing (or imaginary) in the feature, its corresponding feature is termed *open*. That is, as shown in Fig. 7.19 (a), an open bounding feature such as *slots* or *steps* do not bound themselves with the complete ring of the top face (two and three imaginary edges, respectively).



In contrast, a *hole* or a *pocket* in Fig. 7.19 (b) is closed because the edges of an isolated loop does not touch the outermost loop at all and form closed loop. The

dashed line segments shown in above figure are missing lines that originally do not exist in the drawings (no edge is imaginary). Determining the existence of imaginary edges supplies the information whether the feature is open or closed. If the value of *open/closed attribute* is determined as closed, the shape attribute is assigned as '1'. It is '-1' if the characteristics of the attribute is open.



## **Chapter 8**

# **CAD Model Reconstruction and Data Communication**

Once CAD models are recognised from a scanned point set, their location, orientation and dimensions must then be derived. In this research, the geometric information of recognised features is reconstructed from the information of its edge points from the fact that the boundary curves of the feature are corresponded to its edge points.

The aim of this chapter is to explain the mathematical algorithm for feature reconstruction process. In addition to that, since the reconstructed CAD models are required to be exchangeable with other CAD/CAM systems in the manufacturing environment, the standard neutral file formats such as IGES and STEP are also briefly described for data communication.

## **8.1 CAD Model Reconstruction**

CAD model reconstruction is an important issue in itself and is often used as a final stage in the overall goal of 3D-part representation. Hence, a reliable extraction method needs to be employed for reconstructing the dimensions and orientation of recognised CAD models. Although there have been numerous attempts for reconstructing CAD model descriptions to given point data such as iteration methods, the least mean square (LMS) and maximum-likelihood estimation (MLE), the most robust method is the least squares as a MLE theory (Landau, 1987). In this research, therefore, MLE method combining with LMS method was employed to derive the geometric information of recognised features with the information of edge points.

The minimisation of user's intervention in feature reconstruction is crucial for downstream engineering activities such as automated CAPP and CAM. However, reconstructing features is sometimes much easier if the user intervention is allowed at the extraction stage. Hence, the ReTool system provides a user confirmation stage where the user can interactively change the dimension or orientation of a recognised feature with the idea of the parametric design scheme.

In this section, the algorithm of MLE and parametric design scheme are briefly discussed.

## 8.1.1 Maximum-likelihood estimation (MLE) method

### 8.1.1.1 Review of probability concept (Press, 1992)

A reconstruction function needs to be chosen to measure the agreement between the scanned data and the reconstructed *CAD* models with a particular choice of parameter. In this subsection, a brief review of probability concept is discussed since the principle of MLE method is based upon probability concept.

A random variable  $X$  is a function defined on a sample space and the space of possible outcomes of an experiment. A function that determines the probability of each possible outcome of an experiment is called a *probability density function* (PDF) for continuous random variables and *probability mass function* (PMF) for discrete-valued variables. If  $X$  is a continuous random variable, its density function  $f(x)$  must satisfy the conditions

1.  $f(x) \geq 0$  for all  $x$ , and
2. the area under the graph of  $f$  is equal to 1.

The probability that a random variable  $X$  assumes a value in the interval between  $a$  and  $b$ , where  $a < b$ , equals the area under the density function between  $x = a$  and  $x = b$ . The probability that  $X$  will assume a value between  $a$  and  $b$  is given by the



definite integral of  $f(x)$  between  $x = a$  and  $x = b$  and denoted by  $P(a \leq X \leq b)$ , that is

$$P(a \leq X \leq b) = \int_a^b f(x) dx \quad (8.1)$$

The most common continuous valued distribution is the normal distribution. The single variate normal distribution has a probability density function of the following form:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[\frac{-1}{2[(x-\mu)/\sigma]^2}\right] \quad (8.2)$$

The mean and variance of the normal distribution are  $\mu$  and  $\sigma^2$ , respectively, and the standard deviation is  $\sigma$ . Parameters of distributions, such as the PDF functional form, the mean and variance and other parameters can be estimated from samples taken from the distribution. Such samples are called a *statistic*. Since a function of a random variable is a random variable, any statistic is a random variable.

Let  $x_1, x_2, \dots, x_n$  be a sample of  $n$  independent observations taken from the same distribution. The best estimator of the population mean is just the sample mean like below.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (8.3)$$

Good estimator has some of the following properties:

(1) An unbiased estimator is one for which the expected value of the estimator equals the parameter being estimated. The sample mean is an unbiased estimator of the population mean, since

$$E[\bar{x}] = E\left[\frac{1}{n} \sum_{i=1}^n x_i\right] = \frac{1}{n} \sum_{i=1}^n E[x_i] = \frac{1}{n} \sum_{i=1}^n \mu = \mu \quad (8.4)$$

- (2) The probability distribution of a good estimator has minimal dispersion, that is, among all estimators, it should have that smallest or minimum variance.
- (3) Finally, as the sample size of a 'good' statistic is increased, one would expect that the statistic should approach the true parameter value.

#### 8.1.1.2 Feature reconstruction using MLE (Landau, 1987)

Suppose that  $N$  scanned data points  $(x_i, y_i, z_i)$   $i = 1, \dots, N$  are being fitted to a CAD model that has  $M$  adjustable parameters  $a_j$ ,  $j = 1, \dots, M$ . This form of parameter estimation is called as *MLE*. This method presents the likely errors of the best-fit parameters. The objectiveness of this statistical method is to estimate the parameters of the reconstruction function  $a_1 \dots a_M$  that most likely correspond to the input scanned data set in this research.

The model predicts a functional relationship between measured independent and dependent variables,

$$z(x, y) = z(x, y; a_1 \dots a_M) \quad (8.5)$$

Using the least-square method, the reconstruction function for the  $a_j$ 's can be obtained.

$$\text{Minimise over } a_1 \dots a_M : \sum_{i=1}^N [z_i - z(x_i, y_i; a_1, \dots, a_M)]^2 \quad (8.6)$$

As a noise model, assume that each scanned point  $z_i$  has a error that is independently random and distributes as a normal distribution around real model  $z(x, y)$ . It can be assumed that the standard deviations  $\sigma$  of these normal distributions are the same for all points. Thus, the probability of the whole data set is the product of the probabilities of each point.

If the errors are not normally distributed, then the estimations are not maximum likelihood, but may still be useful in a practical sense.

$$P \propto \prod_{i=1}^N \left\{ \exp \left[ -\frac{1}{2} \left( \frac{z_i - z(x_i, y_i)}{\sigma} \right)^2 \right] \Delta z \right\} \quad (8.7)$$

Maximising (8.3) is equivalent to maximising its logarithm, or minimising the negative of its logarithm, hence,

$$\left[ \sum_{i=1}^N \frac{[z_i - z(x_i, y_i)]^2}{2\sigma} \right] - N \log \Delta z \quad (8.8)$$



since  $N$ ,  $\sigma$ , and  $\Delta z$  are all constants, minimising (8.8) is equivalent to minimising (8.6). The accuracy mainly depends on the amount of noise and the size of the model of range data.

Bolle, R. *et al* (1986) give the likelihood estimator for range data to lie on patches of quadric surfaces, such as planes, spheres and cylinders. In *Appendix 5*, the detailed mathematical algorithms of reconstruction processes for plane and cylindrical features are derived as examples.

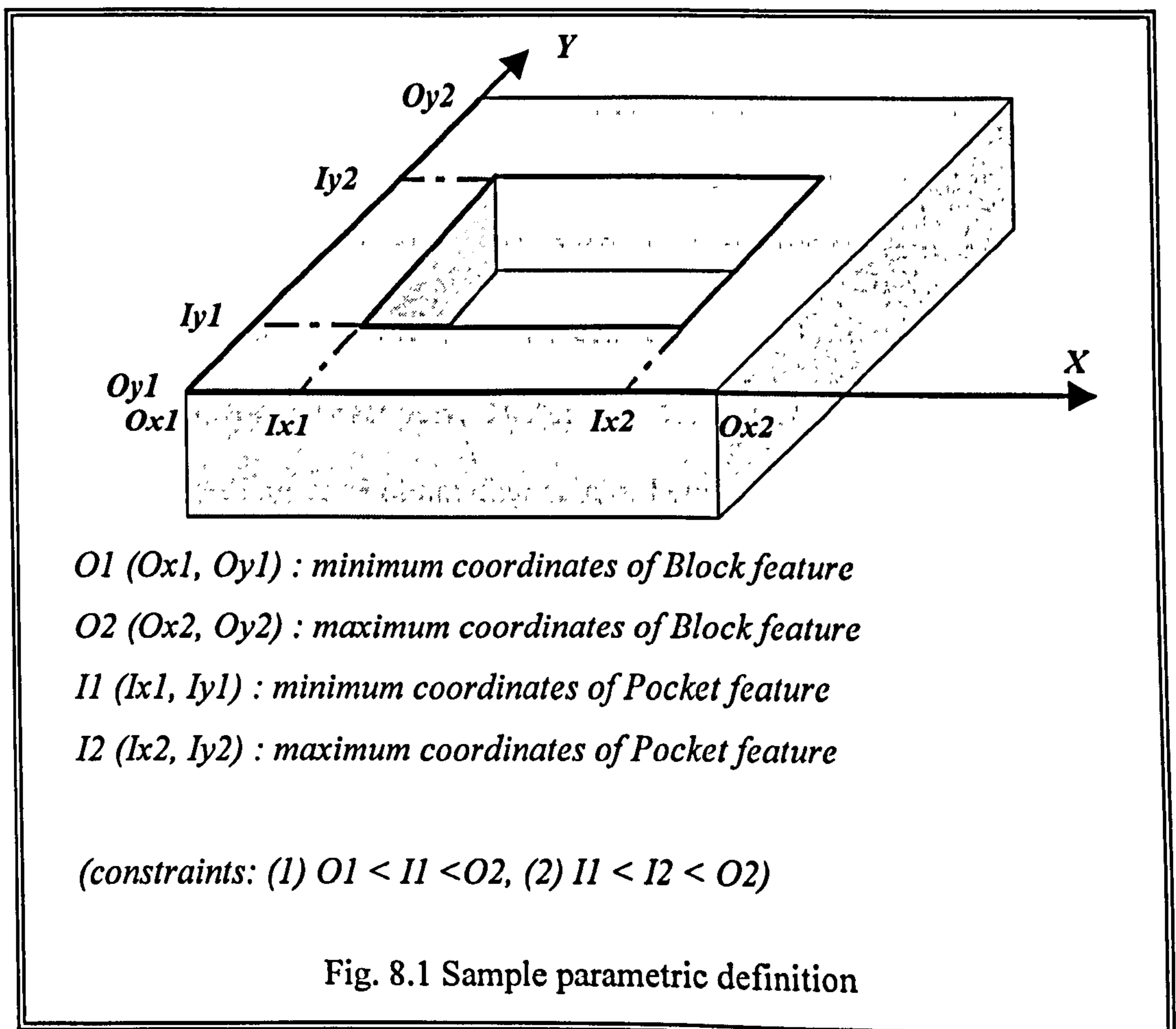
### 8.1.2 Parametric design

The first commercial CAD system developed in the 1970s used explicit modelling techniques. Product design drew the models and gave each part specific dimensions, so that when the model is completed all the parts fitted together (Alschuler, 1988). Explicit modelling techniques remained the state-of-the-art until the first commercial parametric design was introduced in the late 1980s.

*Parametric design (PD)* is an approach that defines the shape of product by using the geometric constraints and the dimension data of topological elements. The geometric constraints describe the relation between the design elements, for example, two elements lie in parallel, they are touching, or they are orthogonal to each other. The dimension data can describe, in addition to the dimensional value of elements, the dimensional relation between the elements by using mathematical equations. PD constructs the required shape by solving the equations that express the dimensional relation between the elements, using the geometric constraints (Toriya, 1993). Therefore PD has the entire model automatically updated if the interrelationship between the various parts of the design changes. The user can add specific key dimensions which the system uses as a basis for configuring the rest of the geometry and determining other dimensions in the design. From that point of view, PD scheme is helpful to modify the existing CAD model to a new model.

In this research, the dimensions of CAD models being reconstructed can be easily changed with the PD scheme. The overall reconstruction process supported by PD is as follows (Shah, 1995).

1. The ReTool system creates the nominal topology of a part model. The result is an individual feature representing the desired geometric elements and their connectivity with default dimensions.
2. Next, if necessary, the user can change the default dimension of model features with the help of geometric constraints. The constraints specify desired mathematical relationships between the numerical variables of the model elements. A shape needs to be modified by only changing the geometric constraints or parameters of the dimensional data specified. An example of 2D geometric constraints for a pocket feature is shown in Fig. 8.1.





Therefore there is no need to modify the geometric shape directly. Designers need only to think about the shape of the product that will satisfy the requirements of the design. Figure 8.2 shows the proposed dialogue box of PD function for an extracted feature.

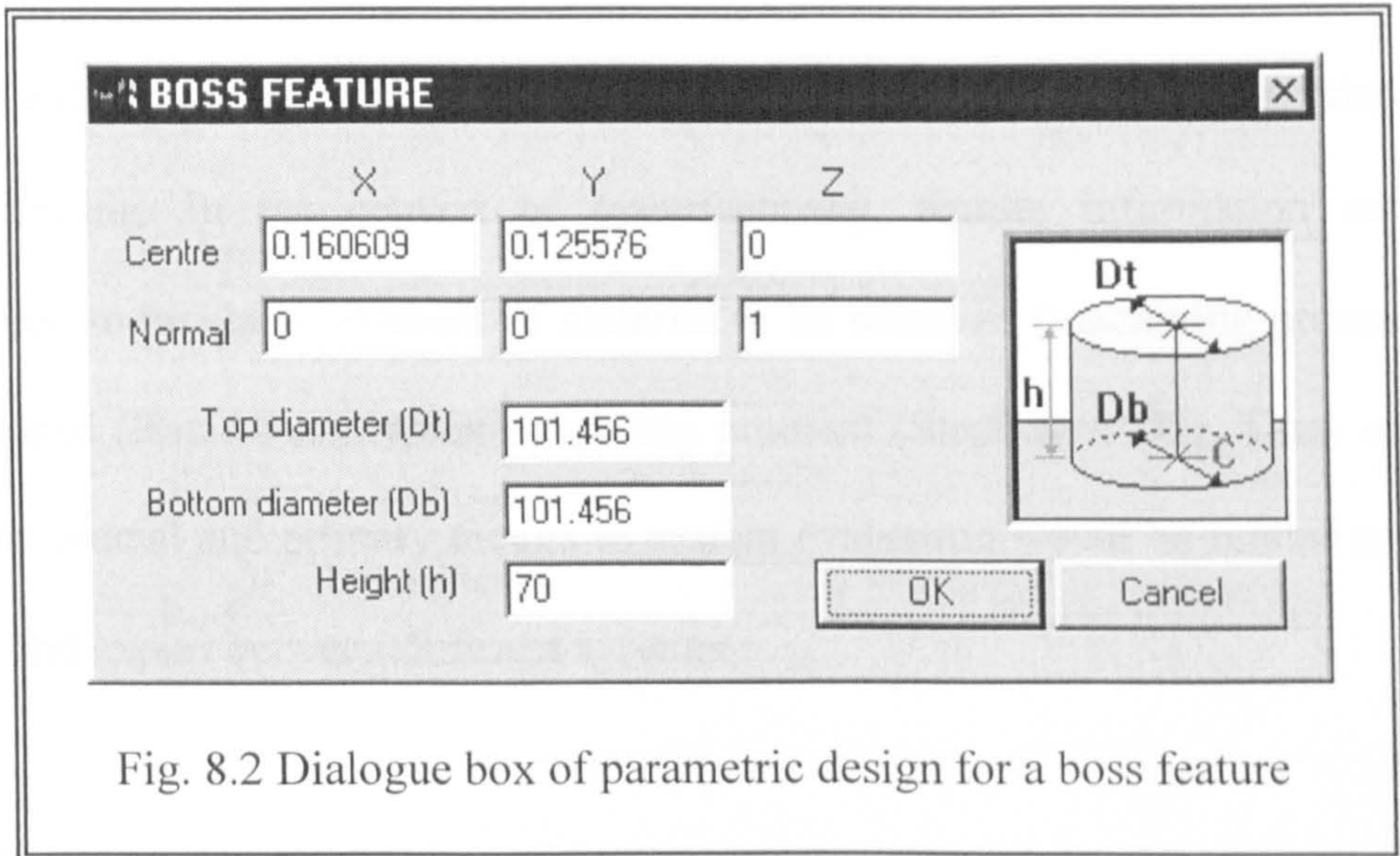


Fig. 8.2 Dialogue box of parametric design for a boss feature

*Centre* and *Normal* fields refer to the centre coordinate of the feature and its orientation vector, respectively. The size of feature can be easily changed by modifying top diameter, bottom diameter, or height box.

After each change, a new instance of the model is created. If the system of constraints cannot be satisfied (e.g.  $height < 0$ ), the system issues a warning message to the user to indicate the nature of the problem. With this proposed application, the dimension of a recognised feature can be changed by modifying any parameter on the dialogue box.



## **8.2 Data Communication**

Recently, there has been a need to share product data between various CAD/CAM applications used during various processes such as design, manufacture, analysis, and planning for reducing the product development cost. For manufacture, RE of mechanical parts must generate complete and accurate part models appropriate for manufacturing. In the context of manufacturing, feature information can be considered to be about volumes of material to be removed (machining process) or to be added (Rapid Prototyping or casting process) (Stephen, 1993). Thus, one of the most crucial and primary factors in system evaluation would be related to data import and export between different systems.

A CAD system usually creates its own internal representation of the part and then converts it to standard file formats for further downstream manufacturing activities. Thus, a standard interface is necessary to convey varied geometric descriptions between different CAD systems. The structure of the data stored in the database of ReTool system was implemented to be compatible with Initial Graphics Exchange Standard (IGES).

The IGES format has been the most commonly used interface standard in industry. It is apparent that IGES provides a means of communicating data between different CAD/CAM systems. However, it has some limitations and imperfections like arrowheads and diameter symbols, which are sometime vanished during transferring them between two CAD systems. The main virtue of IGES is that it has

provided a mean for discussion and a standard for comparison for the more advanced transmission formats that is emerging, such as STEP. The aim of STEP is to produce a single and better standard to cover all aspects of a product life cycle in all industries as an international standard. Although STEP has come out in a state of evolution, many of tools for STEP file are still under evolution.

The descriptions of IGES and STEP neutral file formats are briefly introduced in the following subsections.

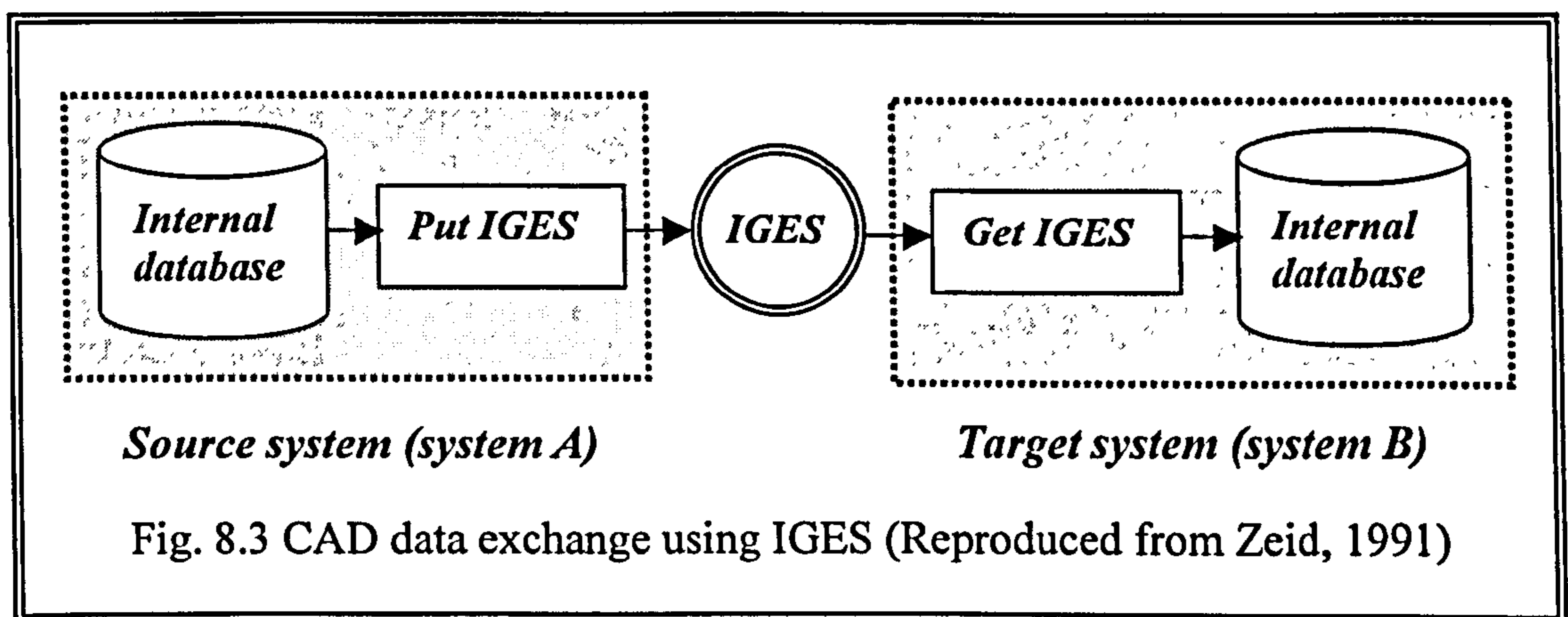
### **8.2.1 An outline of the IGES standard**

IGES is a standard used to exchange product data between different commercial CAD/CAM systems. It was set up as an American National Standard in 1981. IGES defines a neutral data format, which denotes an 'IGES model' of modelling data of a given product. The IGES models can then be read and interpreted by other CAD/CAM systems, so the associating product data can be exchanged between these systems. IGES describes points, line, curves, surfaces and other entities that can be seen in engineering drawings.

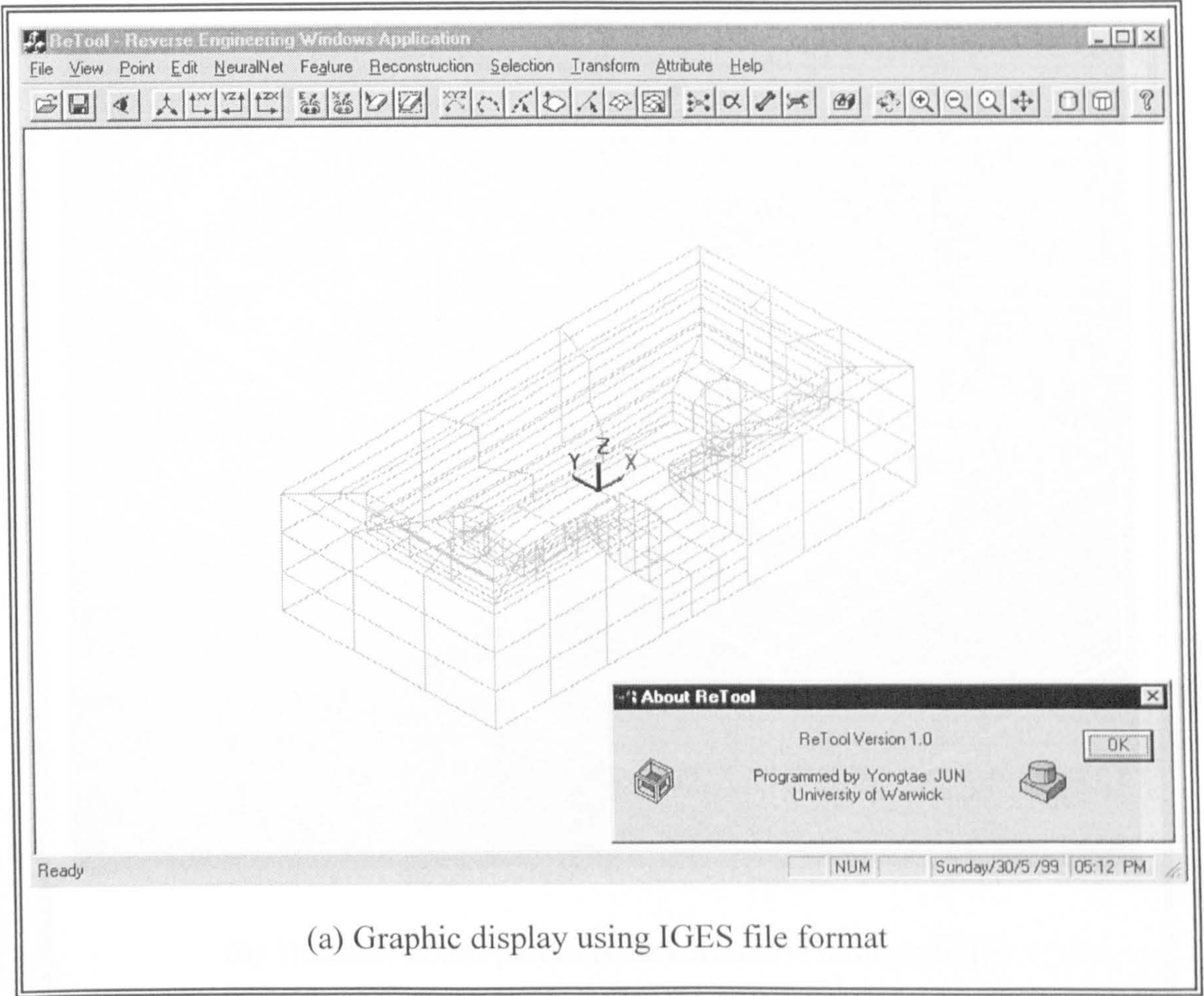
In a CAD model, the fundamental unit of information is the entity. This means that all product definition data are defined as a list of predefined entities. Entities are categorised as geometric and non-geometric. The definition of the product shape and their relationships are represented in geometric entities, whilst non-geometric entities provide views and drawing of the model and include many other characteristics such as, type of dimensions, centre lines, notes, symbols, attributes,



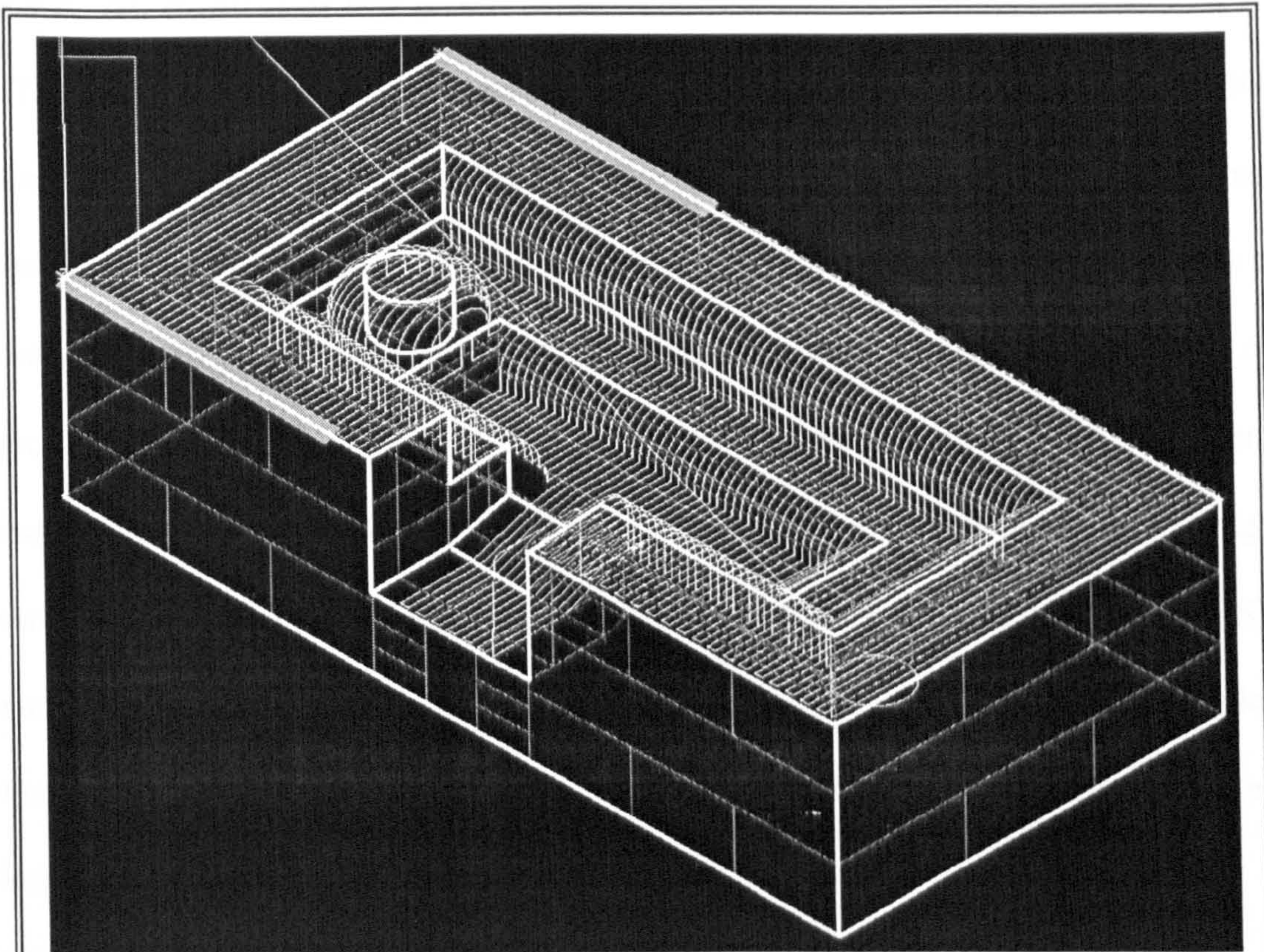
properties and macro. IGES is a useful data exchange format in RE. It only defines entity classes and a sequence number in a neutral format file. For the IGES format, the scanned points must be grouped in sets. Each set of points must be represented as an IGES entity. Using this approach to the data format gives more flexibility to a different kind of commercial applications. Figure 8.3 shows the data exchange using IGES.



One of the main drawbacks in using IGES processors is the precise mapping of internal entities of a given CAD/CAM system to IGES entities and vice versa. If the mapping is not carried out precisely, loss of shape, accuracy and data of the entities may occur and finally result in the failure of the exchange process. Especially, the major drawback occurs whenever an entity is not specifically supported and thus must be transformed to the closest available entity between two different systems. A detailed example of IGES entities concerning B-spline surfaces is given in Appendix 6. Figure 8.4 (a) shows the original shape of a test part reconstructed from the ReTool system, and it is exported with IGES file format. *CADDS5<sup>TM</sup>* imported the IGES file and its visualisation is shown in Fig. 8.4 (b).







(b) Transferred test part into the *CADD5<sup>TM</sup>* through IGES

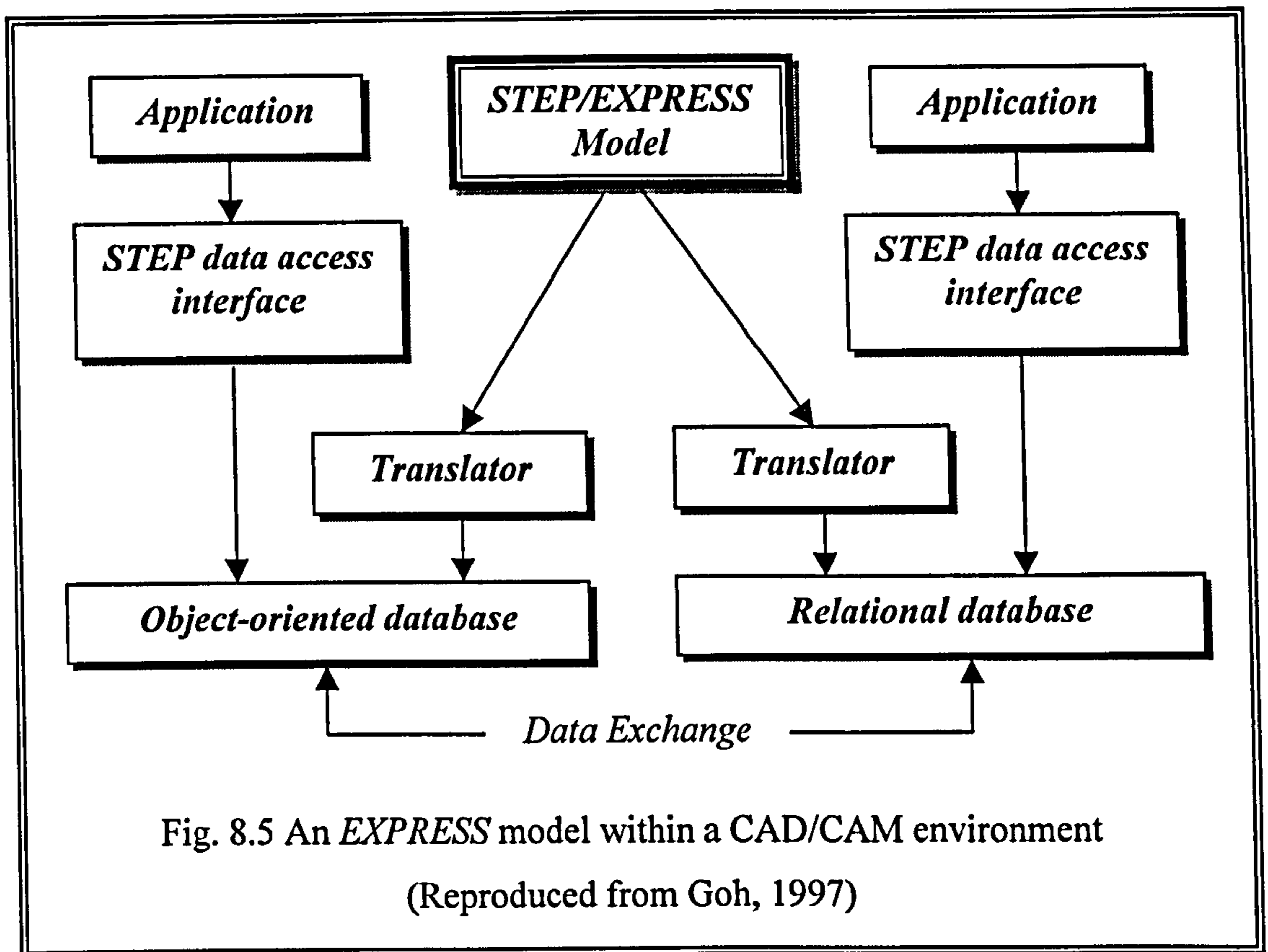
Fig. 8.4 Export the test part to commercial CAM system using IGES



### 8.2.2 An outline of the STEP standard

STEP standard is a further development that is assumed to become the ISO standard for graphic file transfer. STEP includes representation of product data and the mechanisms necessary for the exchange of product data. The STEP effort is a multi-national endeavour to create a standard to support sharing and exchange of product data. STEP is valuable for product databases, data exchange and concurrent engineering (Warthen, 1992). Each part of the STEP standard needs go through three major development phase: Draft (CD), Draft International Standard (DIS), and International Standard (IS). STEP consists of a set of fundamental principles that apply to all parts, a definition language (*EXPRESS*), a set of Integrated Resources (IR), and Application Protocols (AP).

STEP files for a part is defined by the *EXPRESS* language. Its purpose supports precise definition of STEP information models. It not only defines entities that describes the part, but also specifies constraints on attributes, definition of schemas, data types, rules, functions and procedures. It also supports multiple inheritance and partial binding of methods to entities. Part database can be defined and manipulated using *EXPRESS* as shown in Fig. 8.5.



*EXPRESS* is not a programming language. There is no compiler and executable code for *EXPRESS*. The language itself resembles a mixture of different languages including *ADA*, *C*, *C++* and *SQL* (Goh, 1997). As it is basically a data definition language, it does not contain any elements which support data manipulation in the form of input, output or data processing. Instead, it has a set of *constructs* to enable an unambiguous object definition. *EXPRESS* is understandable by human and computers. *EXPRESS* enables a schema to be declared. The schema consists of entities which are related parts of interest and data types to support the definitions of these entities. *EXPRESS* also supports function, procedures and global rules.

STEP documents are organised into parts, referred to as ISO 10303- $x$ , where  $x$  is the part number.

- Series 1      *Overview of STEP*
- Series 10     *EXPRESS documents*
- Series 20     *Implementation documents*
- Series 30     *Conformance documents*
- Series 40     *Generic Integrated Resources documents*
- Series 100    *Application Integrated Resources documents*
- Series 200    *Application Protocol*

Part 48 defines a form feature to be a shape aspect that conforms to some preconceived patterns. A shape aspect is defined as a shape. The specification of a feature consists of a feature name, one or more parameters, and an optional descriptive label. Part 48 deals with feature information at two levels, listed below:

- *Form feature schema.* Consists of generic shape properties, with no application connotations and no specification of representation.
- *Form feature representation schema.* Consists of a variety of representations to model shape properties for use in geometric modelling.



Part 48 schema supports three types of features: volume features, transition features and feature patterns, each of which has several subtypes, as shown in Fig. 8.6.

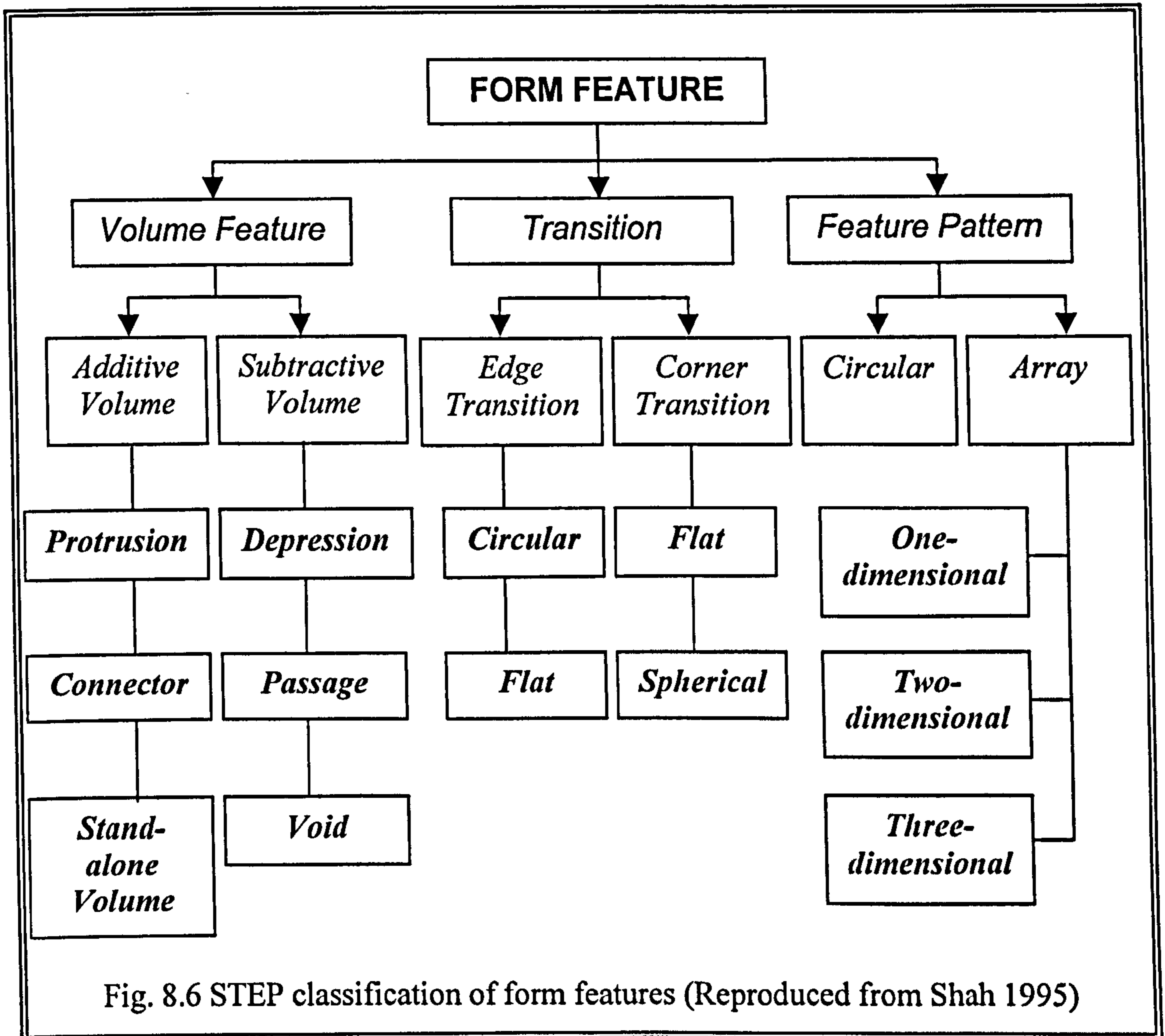


Fig. 8.6 STEP classification of form features (Reproduced from Shah 1995)

As shown in Fig. 8.7, volume feature can further be divided into six instances that are commonly used in CAD application such as passage, depression, protrusion, stand-alone, connectors and void.

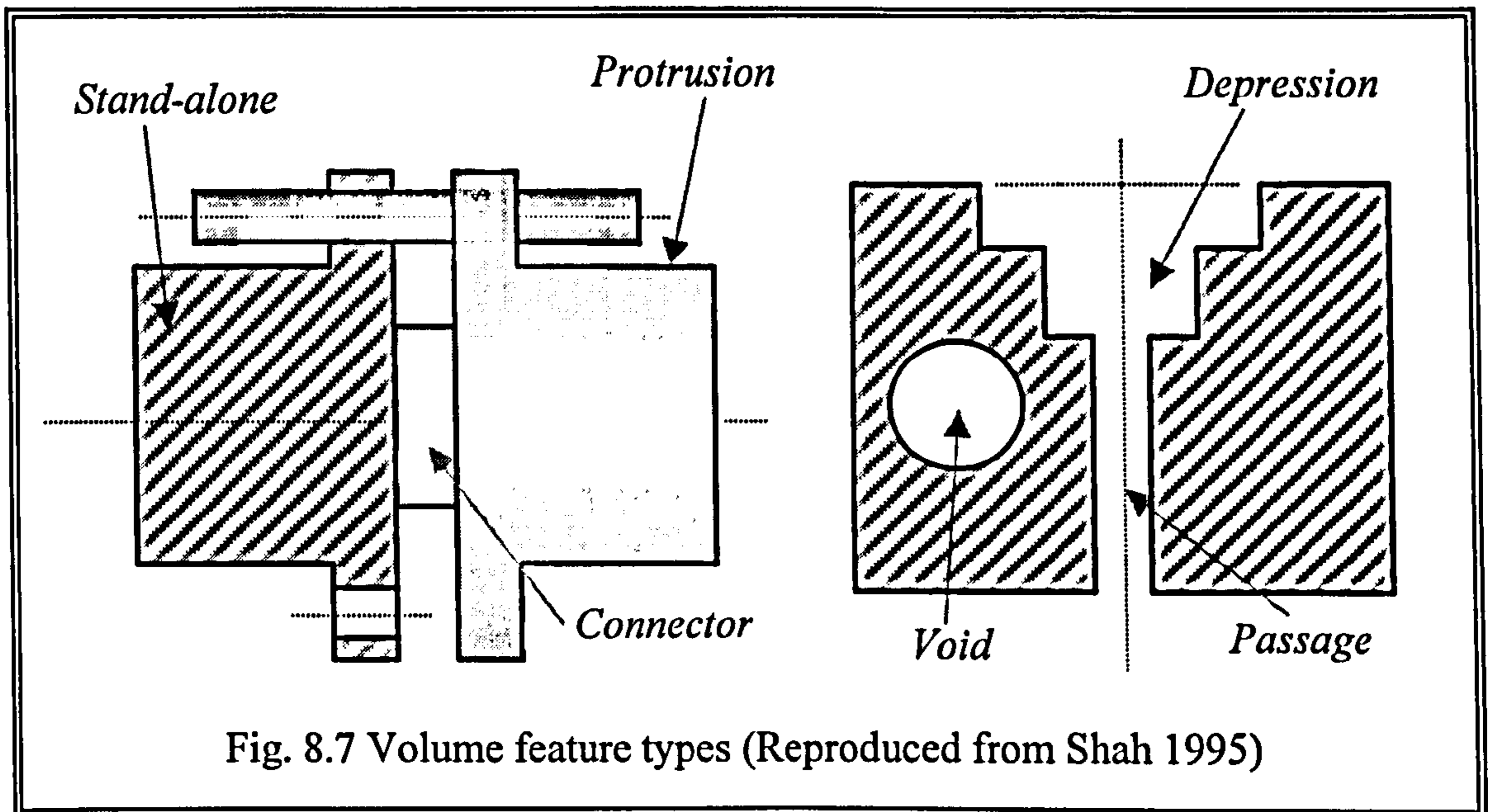


Fig. 8.7 Volume feature types (Reproduced from Shah 1995)

- *Passages*: subtracted volumes that intersect the pre-existing shape at both ends.
- *Depressions*: subtracted volumes that intersect the pre-existing shape at one end.
- *Protrusions*: added volumes that intersect the pre-existing shape at one end.
- *Stand-alone*: additive volume that does not connect previously defined volumes of the part.
- *Connectors*: additive volume that connects two pre-existing shapes at both end.
- *Void*: subtractive volume completely enclosed by pre-existing volume.

## Chapter 9

# System Implementation and Examples

The overall architecture of the ReTool system and a typical example of reconstructing CAD models are discussed in this chapter.

### 9.1 System Implementation

The prototype of a feature-based RE system, ReTool, has been developed to implement the concept of feature recognition using ANNs. The system was implemented using C/C++ language. The user interface and the visualisation of the model geometry were implemented with Visual C++<sup>TM</sup> MFC and OpenGL<sup>TM</sup>. For the internal geometry representation, an application program interface (API) kernel was written and used by the author. The modularity of the ReTool system architecture allows the use of standard software component. The functional architecture of the system is depicted in Fig. 9.1.



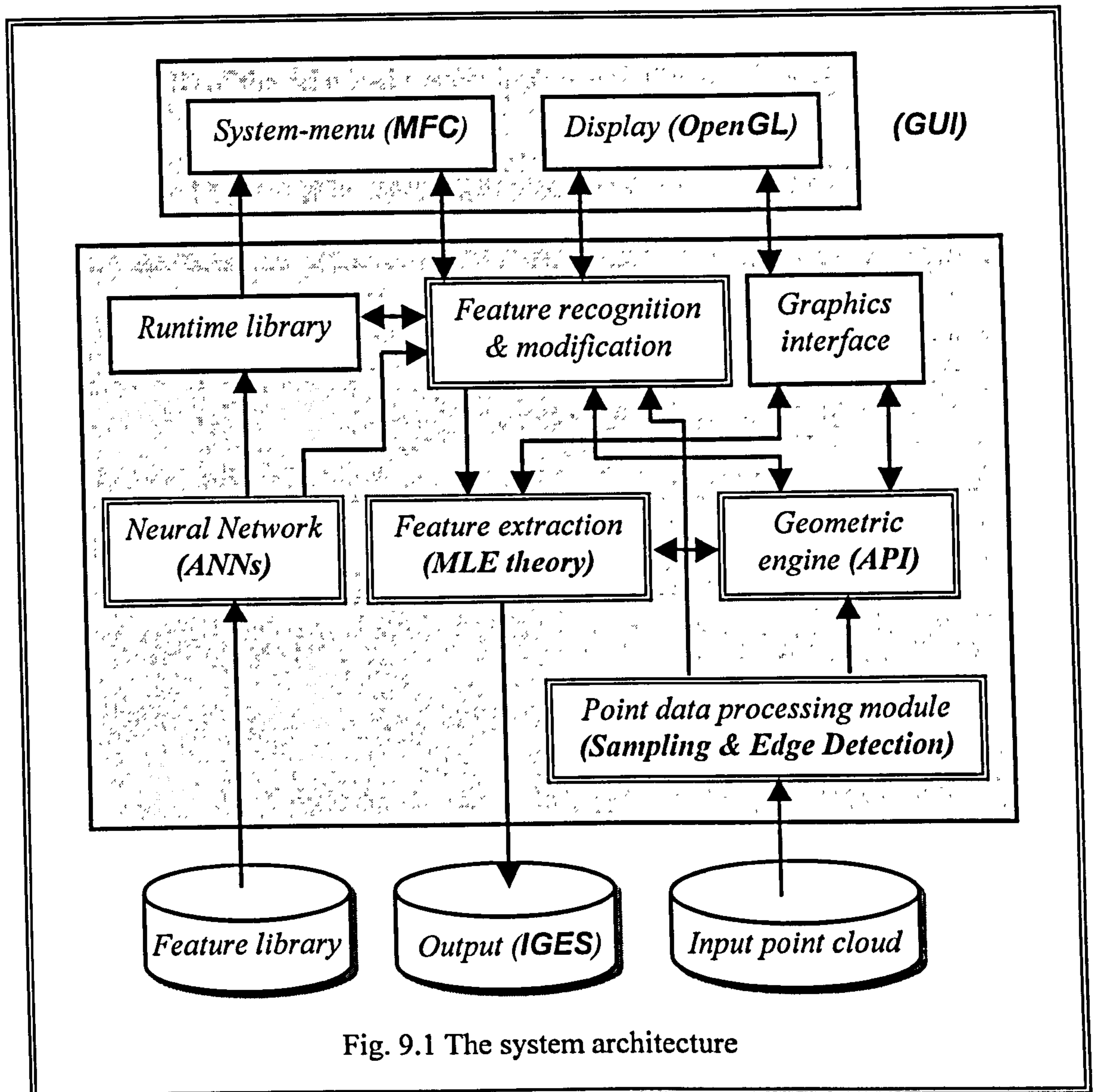


Fig. 9.1 The system architecture

ReTool system provides a highly intuitive user interface and no special CAD background is required. All functions are activated by pull-down menus, tool bars, and dialogue boxes. Visualisation is one of the most crucial factors in CAD systems. The system offers powerful shading function and transformation functions such as zoom, translation, and rotation. Area segmentation and entity selection is

very straightforward by just selecting or clicking the mouse. The graphical user interface (GUI) of the proposed system is shown in Fig. 9.2

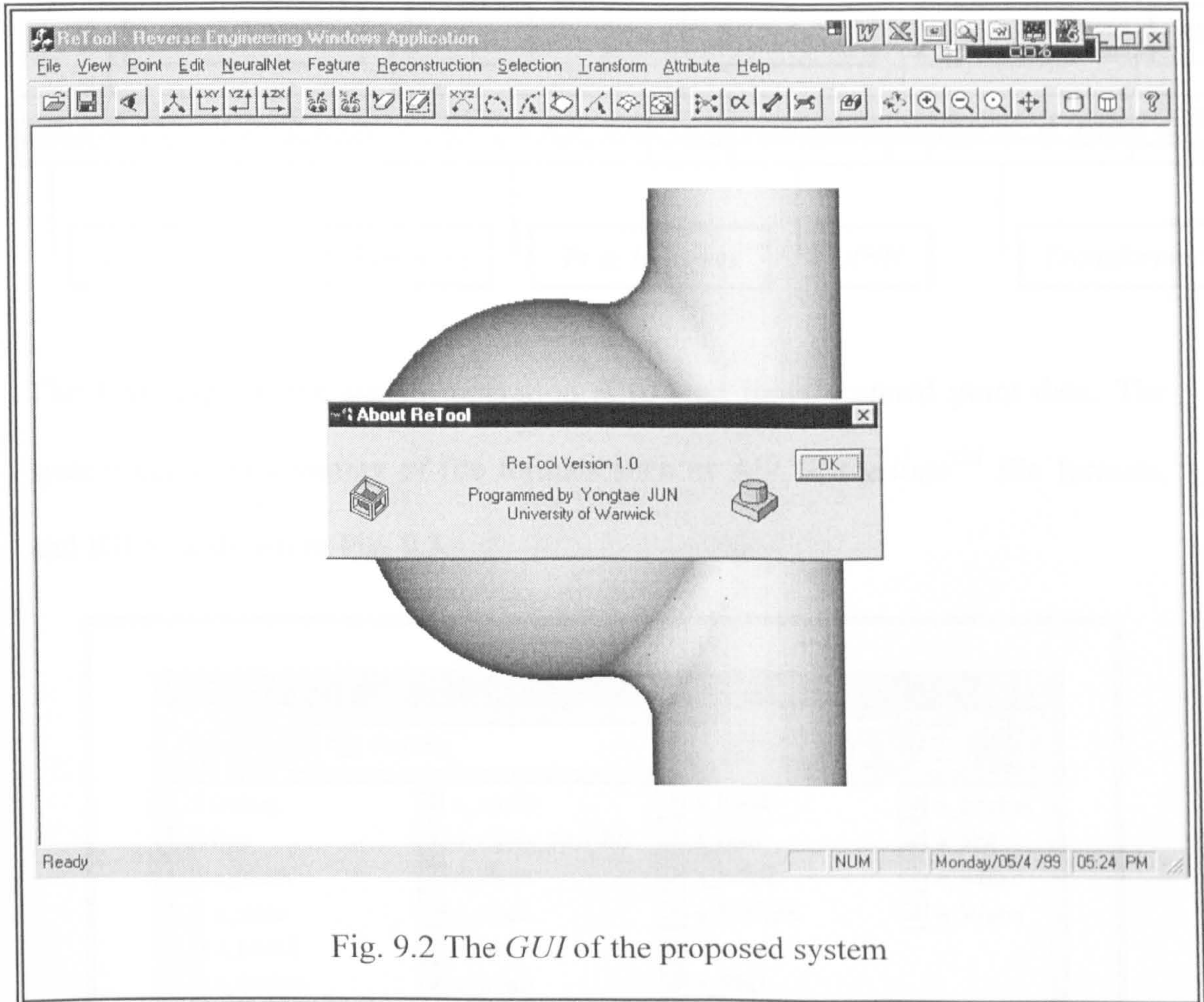
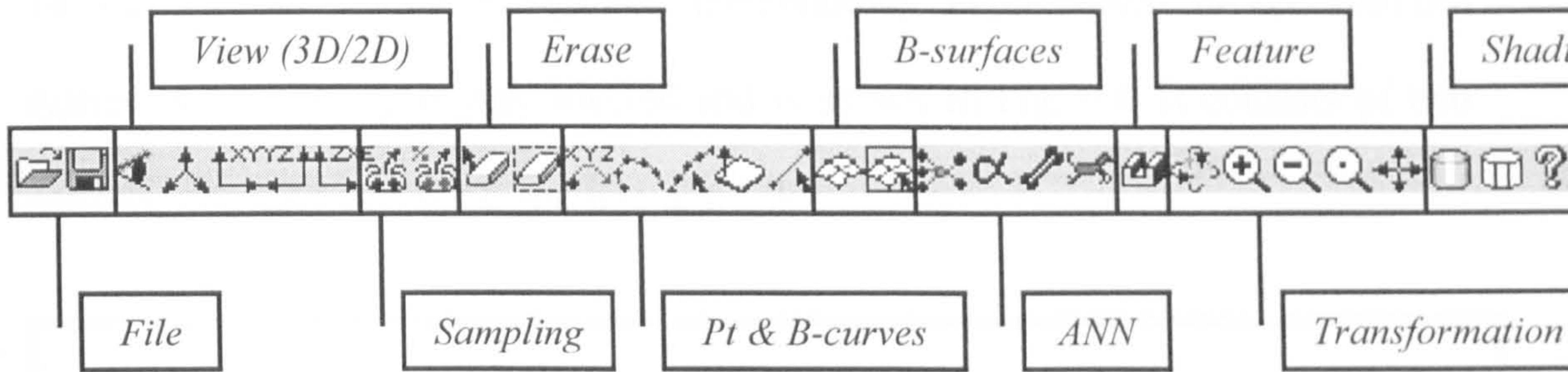


Fig. 9.2 The GUI of the proposed system



The system provides a set of toolbars as follows. Each toolbar corresponds to one of following main functions in the system menu.



The first stage of the system operation is to read input scanned point data. The system can read a variety of file formats such as ASCII, *Cyclone*<sup>TM</sup> file formats, and IGES as shown in Fig. 9.3.

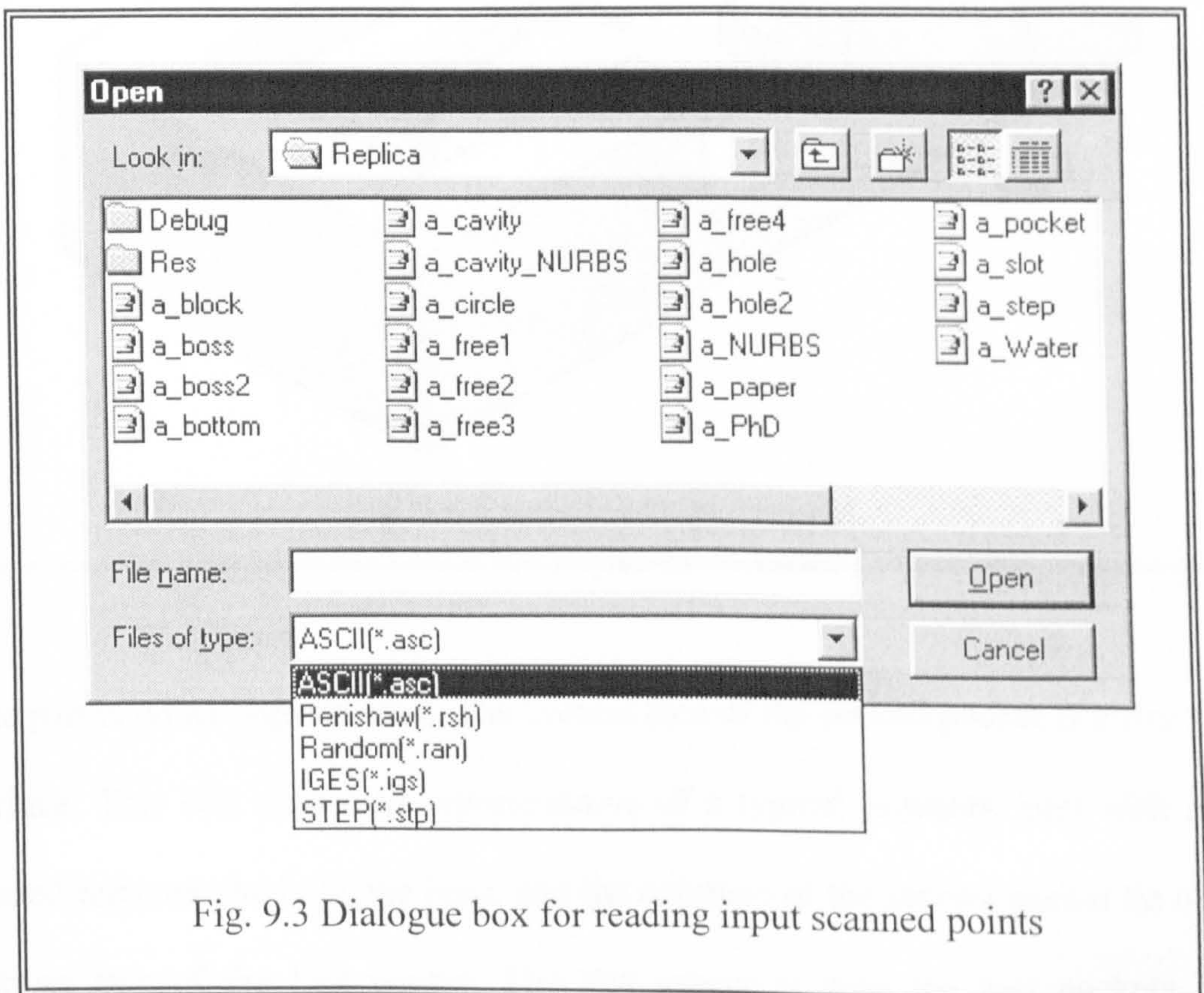


Fig. 9.3 Dialogue box for reading input scanned points



## 9.2 Examples

To validate the feature recognition methodology implemented in the ReTool system, a prismatic part was selected and is shown in Fig. 9.4. It consists of two pockets, one hole, one boss, and one blind slot.

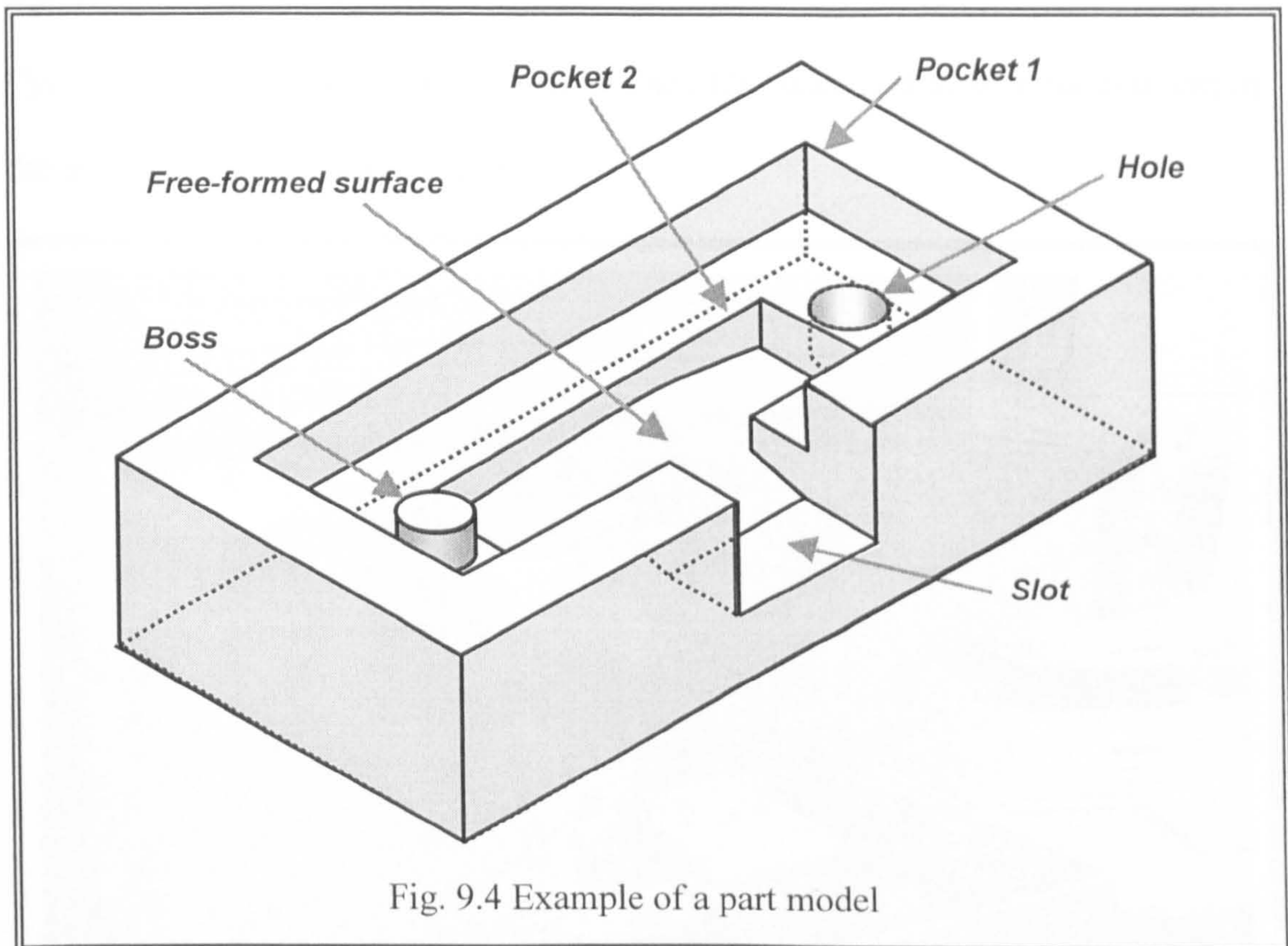


Fig. 9.4 Example of a part model

The part is rather sophisticated. The bottom face of the second pocket is a free form surface. This was chosen as representative of a typical prismatic part with inter-related features: the hole, the boss, and the entrance of the second pocket lie on the bottom face of the first pocket. The slot intersects with the two pockets. This



example part provides an adequate usability of the ReTool system. The part is approximately  $100\text{ mm} \times 50\text{ mm} \times 24\text{ mm}$ . The whole process with scanning, point data processing, segmentation, recognising and reconstructing of the part is described in the following subsections.

### 9.2.1 Point data capture

*Cyclone*, a scanning device by *Renishaw, Inc., UK*, was used to scan the test part in the experiment as shown in Fig. 9.5.

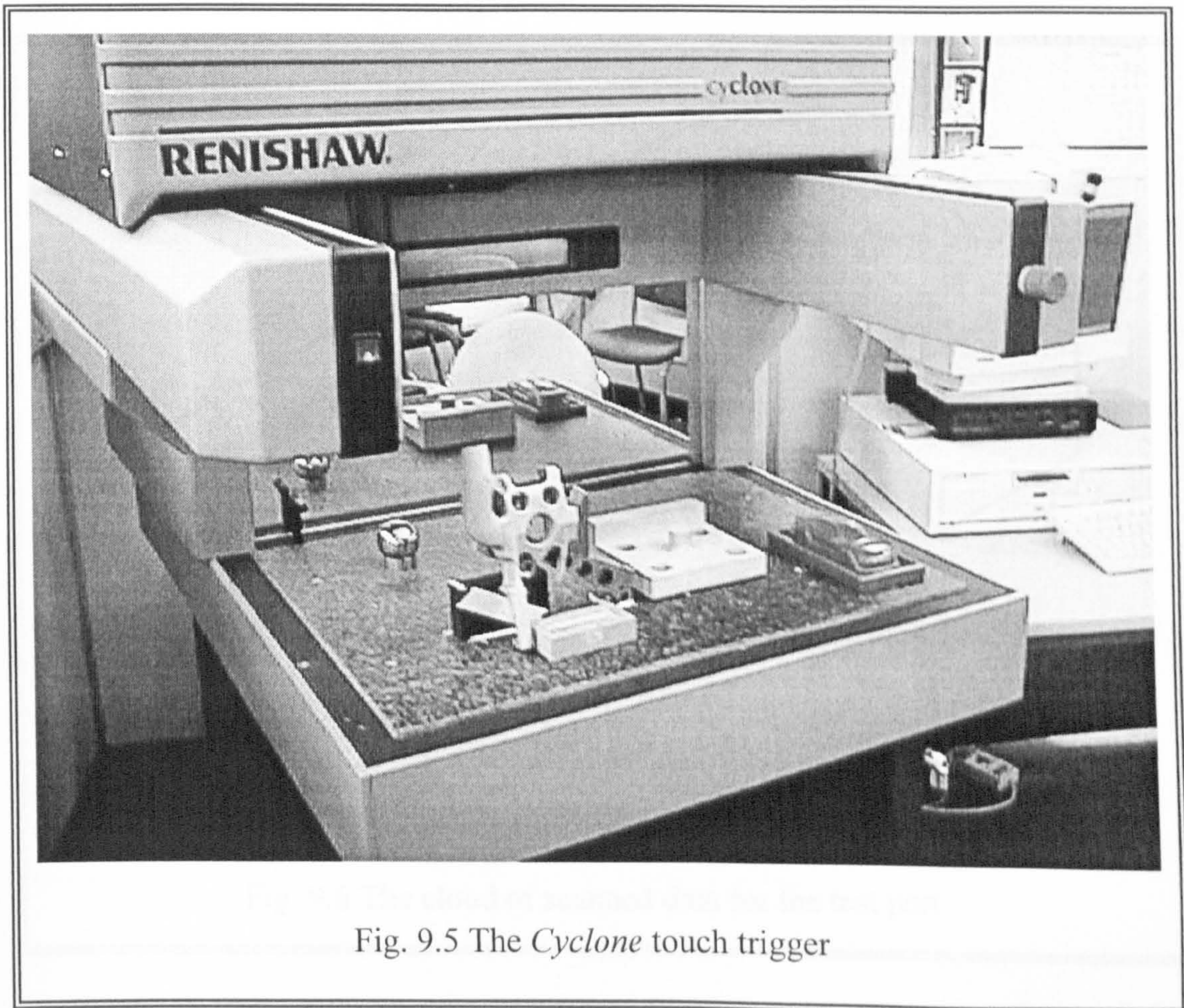


Fig. 9.5 The *Cyclone* touch trigger



The *Cyclone* machine is a dedicated stand alone unit, designed specifically for high speed data capture of complex 2D and 3D forms. It has a nominal measurement accuracy  $\pm 50$  microns under optimal conditions, but it depends on the nature and shape of a part surface in practice. The system uses *Renishaw's* own PC-based scanning software *Tracecut<sup>TM</sup>*, to control the scanning operation. The software internally accomplishes the compensation of the probe radius. Thus, the scanned data was used to directly reconstruct CAD models. The scanned point clouds are shown in Fig. 9.6.

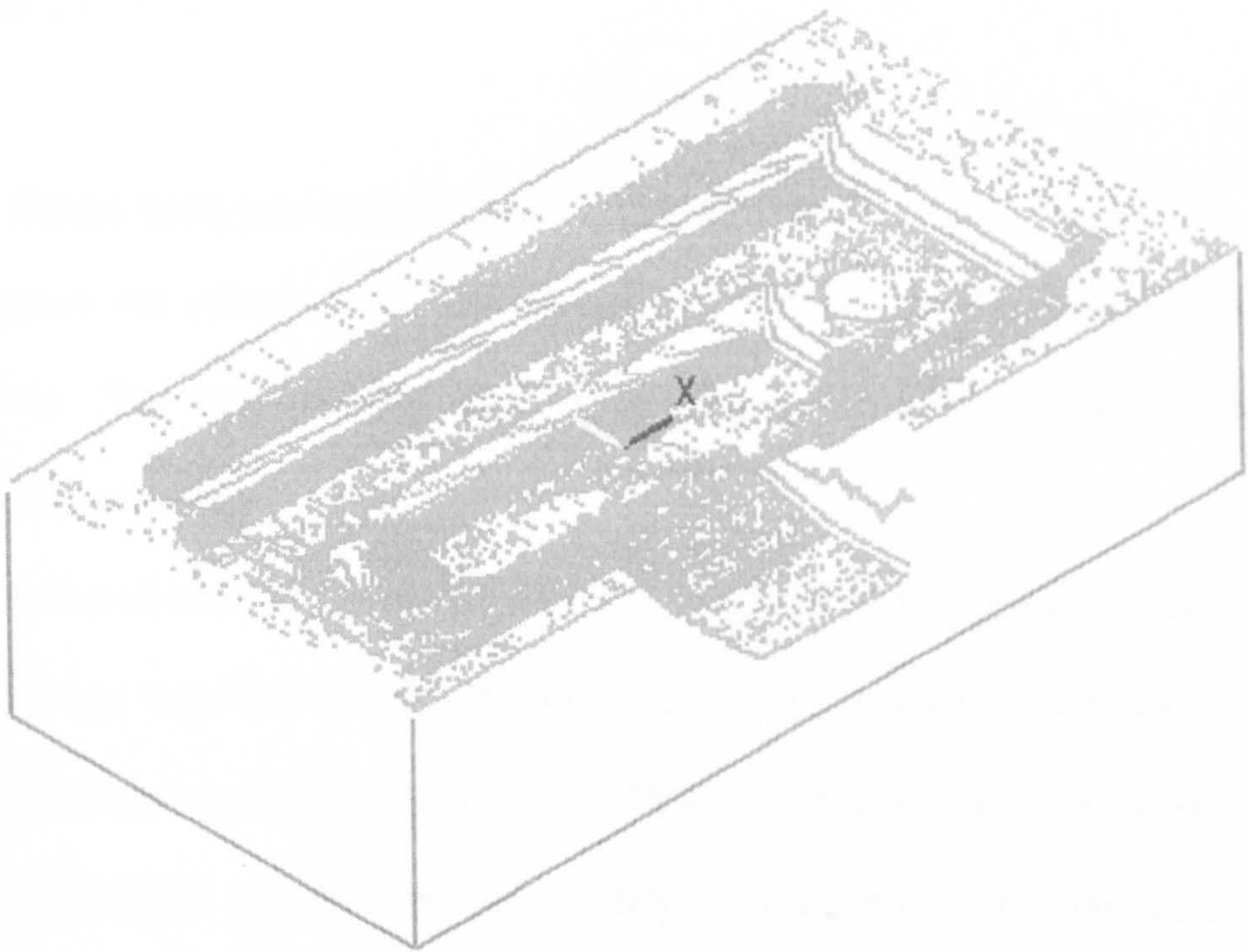


Fig. 9.6 The cloud of scanned data for the test part



The probe is a pressure-sensing device that is triggered by any contact with a surface. The captured points are structured. The scanning process was undertaken in a raster scan mode. That is, the scanning probe moved horizontally along one direction, say  $y$ , and then stepped a path interval along to  $x$ -direction. It moved at a constant speed and recorded the data continuously. The step size (tolerance) and the path interval were maintained constant during the whole scanning process. It automatically follows the surface of a physical part, then the linear distances moved along the three axis are recorded by length-measuring systems, thus providing the  $x$ ,  $y$ ,  $z$  coordinates of the point (ASCII data). The total number of scanned points acquired was 33165.

### 9.2.2 Point data processing and edge detection

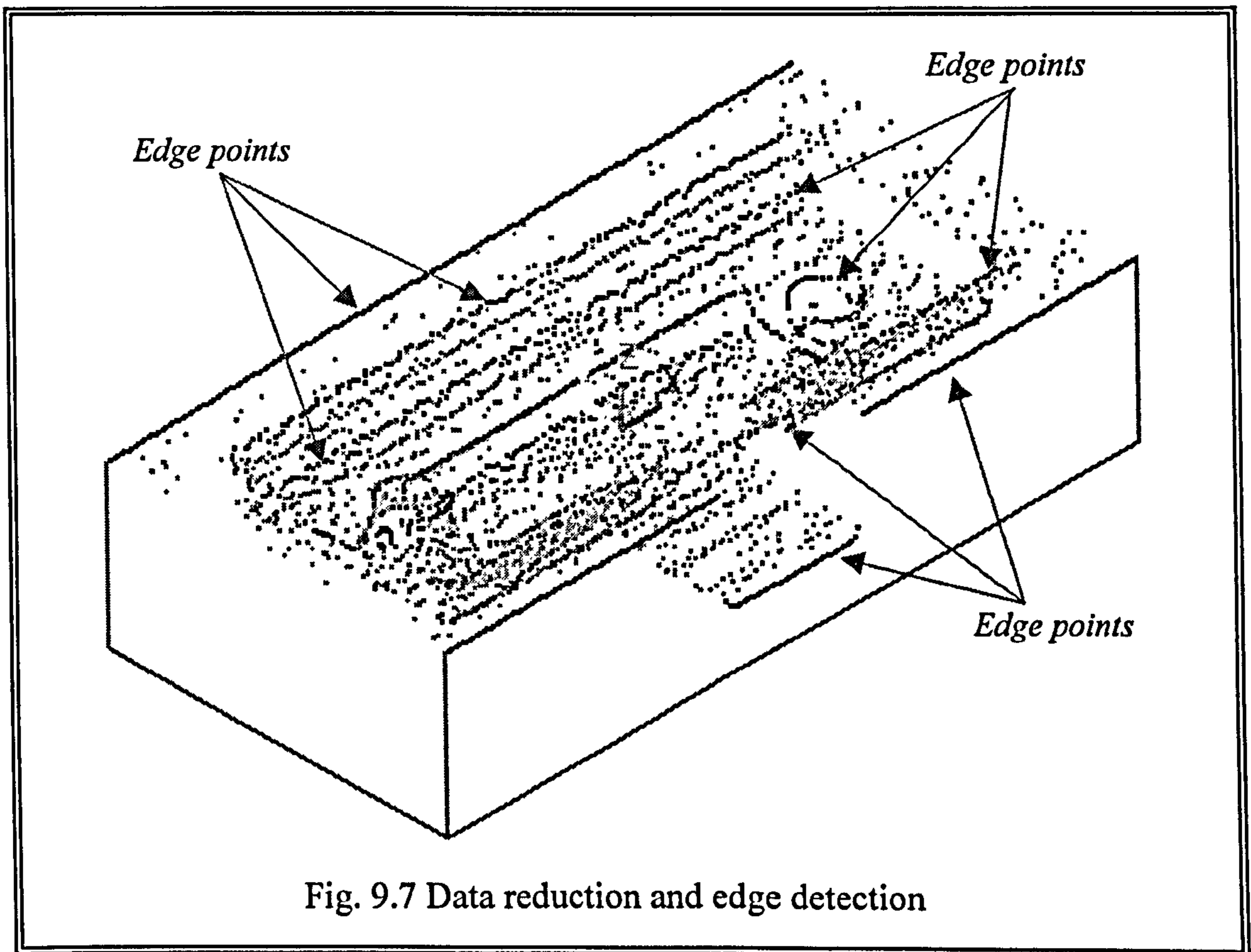
To improve the efficiency of RE process, the scanned data set needs to be pre-processed. The sampling and edge detection are two crucial steps in the pre-processing procedure.

Two developed sampling modules were applied to the initial scanned point set with the following sequence. Firstly, the sampling algorithm using *directional vector set* was applied into the initial scanned data. Then the scanned data was sampled again by *the algorithm of sampling rate*. These two sampling modules reduced the scanned data set considerably within pre-defined tolerance.

In this example, the sampling rate was input to '2', which means that half of the number of original point data was sampled. This is because every point which

sequence number was multiple of '2' along a scan line was removed. Edge points were simultaneously searched with these sampling processes, so they can be kept regardless of the sampling jobs.

As shown in Fig. 9.7, detected edge points are highlighted (bold) with black colour on the displayed monitor.



It can be seen that edge points were readily detected by the proposed method, which investigate tangent vector set along to scan lines. Once the same process is also applied to the direction that is vertical to the direction of original scan line, all edge points may be uniquely detected. Since scan line paths were set parallel to the  $y$ - $z$

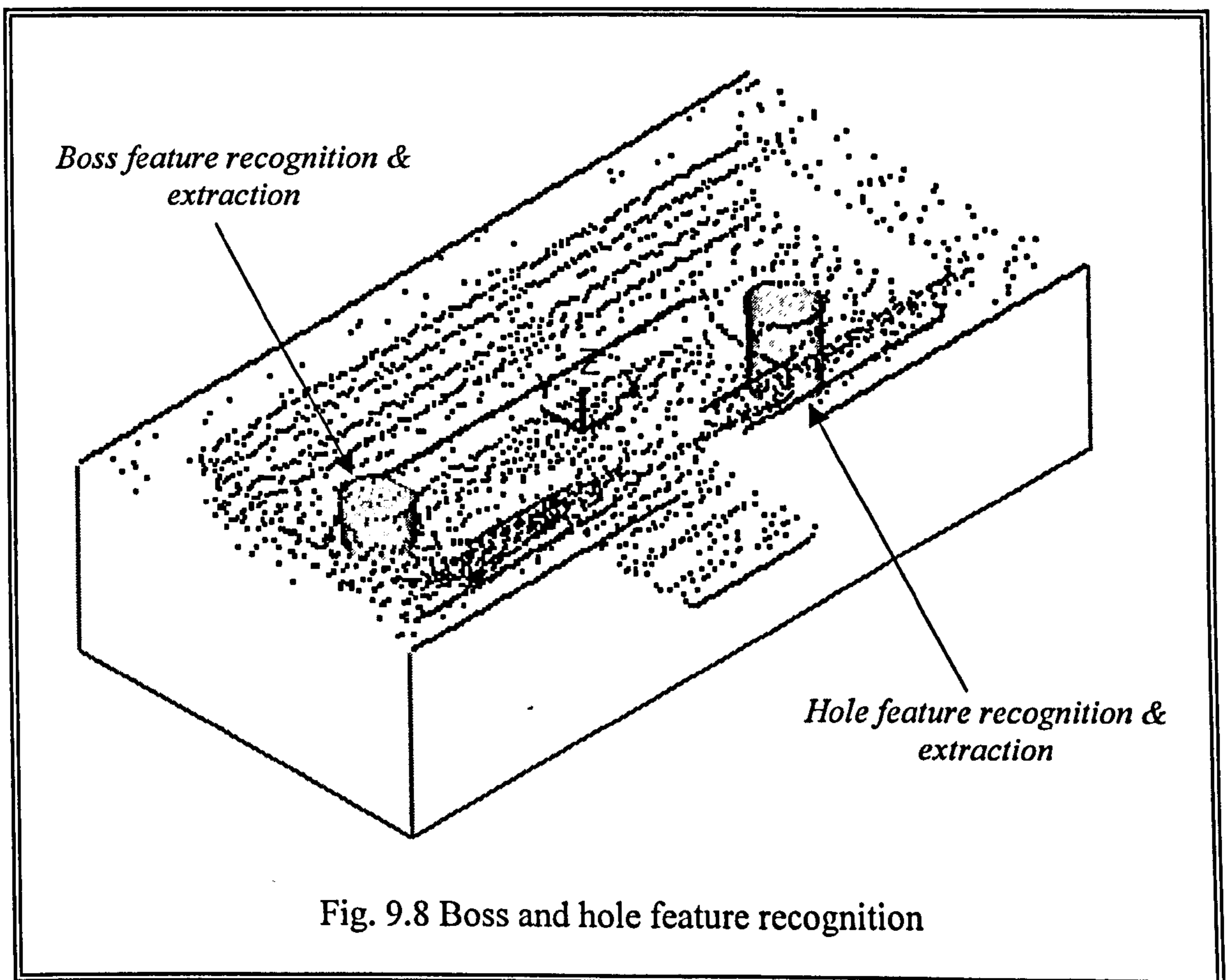
plane, points on the same scan line have the same  $x$  coordinate. This means that any vector manipulation for edge detection and sampling on the  $y$ - $z$  plane can be computed.

If an input scanned point set is unstructured, the sampling algorithm by voxels can be applied. The length of cube can be changed with the tolerance of associated scanner, so that all point in a voxel are discarded except one scanned point per voxel.



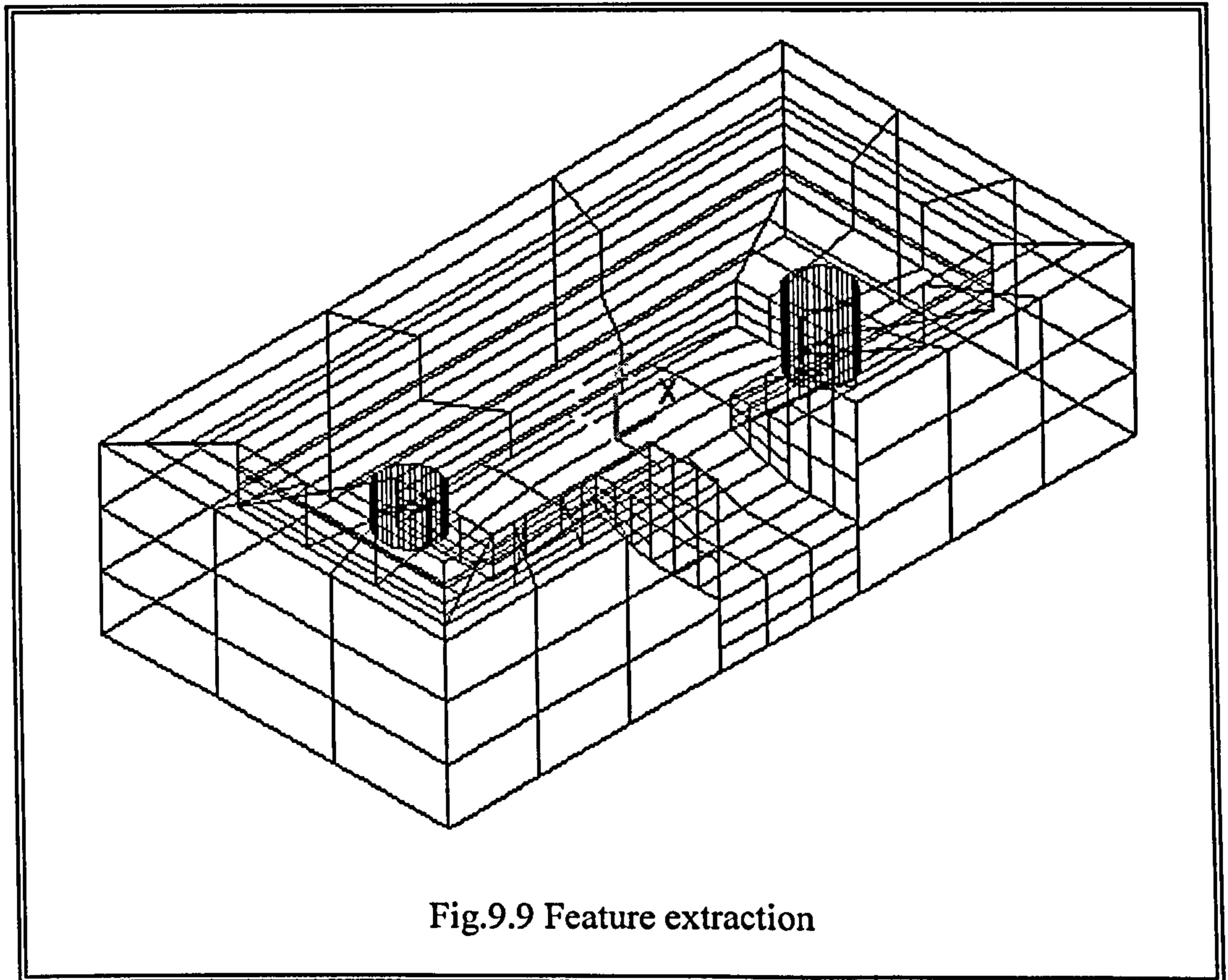
### 9.2.3 Feature recognition

In order to recognise a feature, the geometric attributes of each segmented point set were computed: *chain codes*, *convex/concave*, *circular/rectangular*, and *open/closed*. Then the derived geometric attributes were input to the ANN module. All features were successfully recognised by the ANN. Figure 9.8 shows two cylindrical features are recognised and extracted.



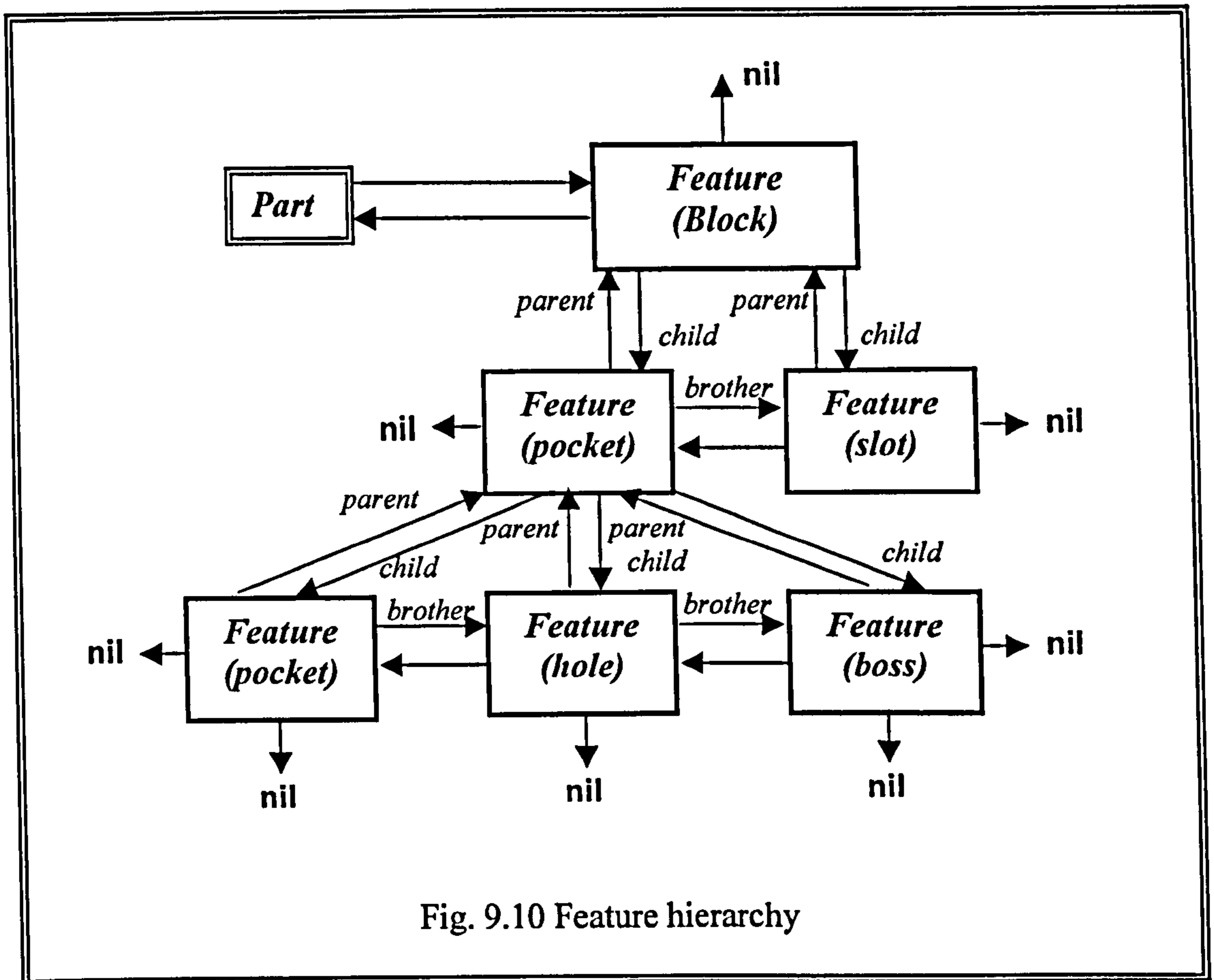
### 9.2.4 Feature reconstruction

The edge points were used to extract the dimensions of recognised feature such as length, depth, diameter, and its orientation. Each individual feature is reconstructed using the feature reconstruction modules as shown in Fig. 9.9.



The extracted features were transformed to B-rep solid models bounded by B-spline surfaces. They can also be transferred into other commercial CAD/CAM systems using IGES file structure for further engineering activities.

The constructed feature hierarchy is schematically shown in Fig. 9.10.



The multiple features of the test part were organised into a hierarchy where the nodes represent instances of the basic features and links represent the parent-child and brother relations.

The link between the nodes provides information on the order of the features. Due to the fact that the reconstructed part geometry depends on the evaluation sequence, every evaluation of the feature hierarchy is performed bottom-up beginning on the left side.



### 9.2.5 Manufacturing and error analysis

After reconstructing CAD models, the next step is to manufacture a replica using the information of reconstructed CAD models. ReTool system enables the reconstructed features to be exchangeable with commercial CAM systems by providing a means of communicating data through IGES standard file format. The IGES file of the reconstructed part was transferred to *CADDS5<sup>TM</sup>*, and its tool path was generated. As shown in Fig. 9.11, to verify the generated tool paths, *VERICUT<sup>TM</sup>* was used.

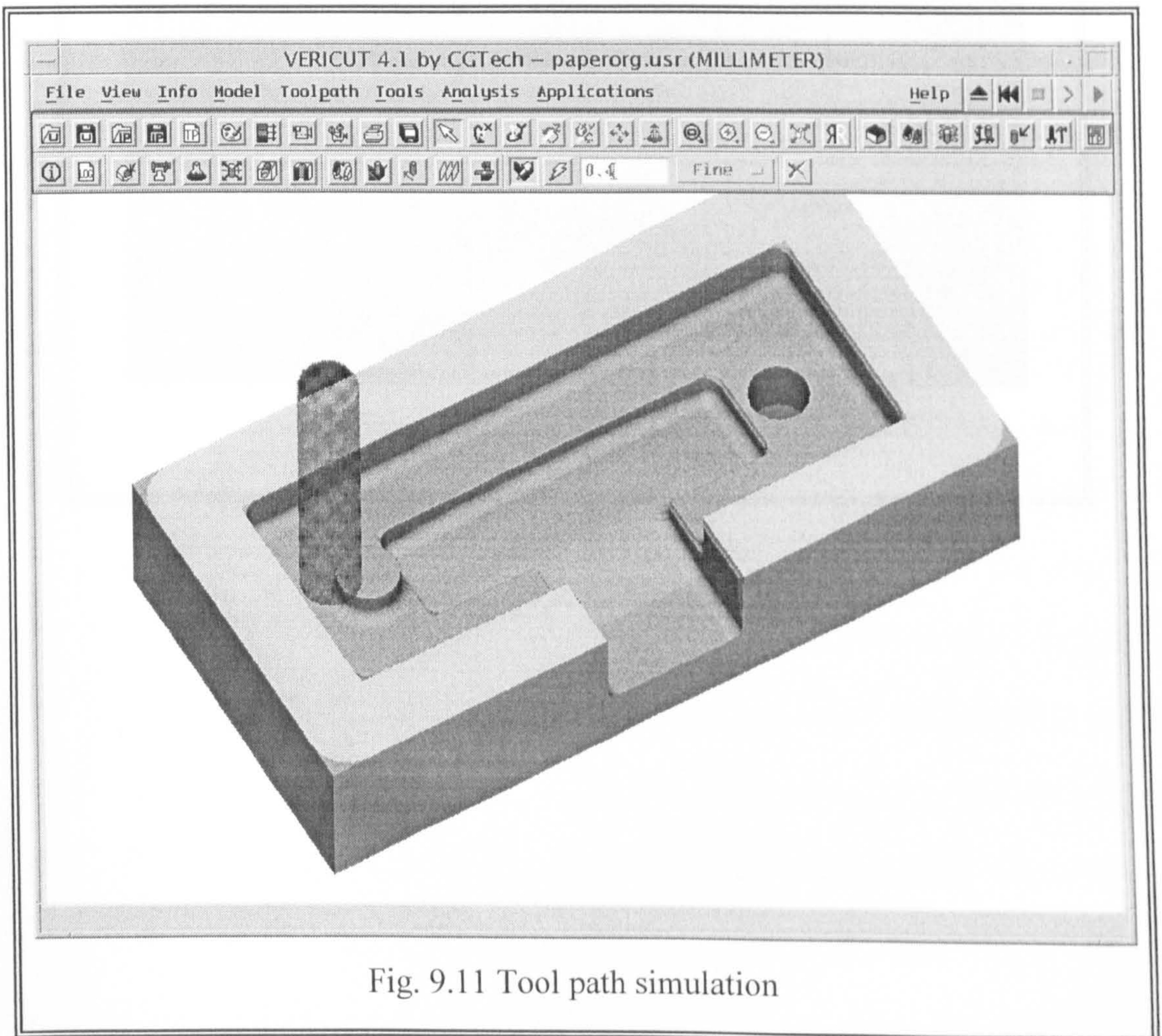


Fig. 9.11 Tool path simulation



Instances of the test part were carefully machined using a 3-axis NC mill, a *Bridgeport™ machining centre, VMC 1000<sub>22</sub> digital*, as shown in Fig. 9.12.

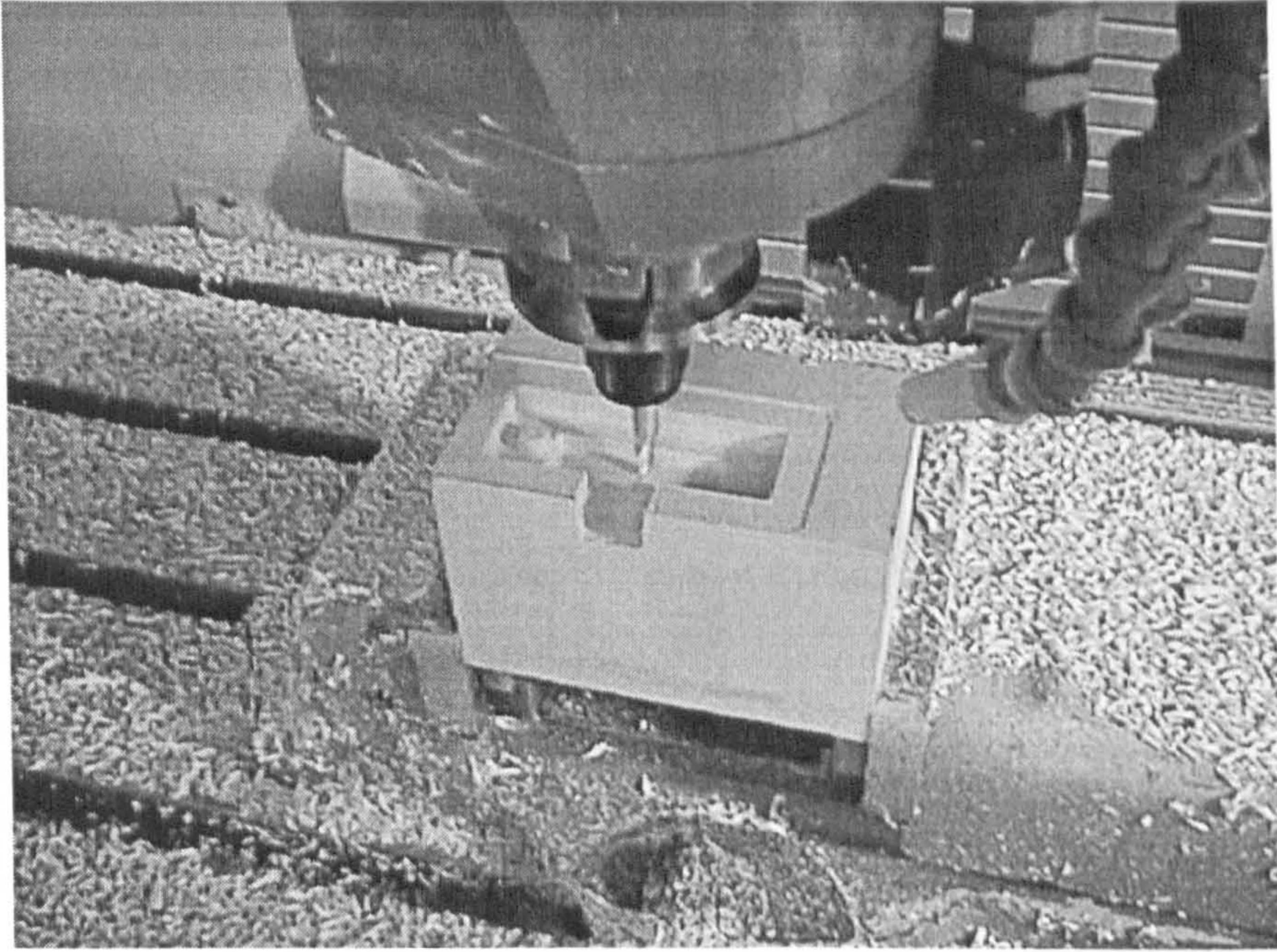
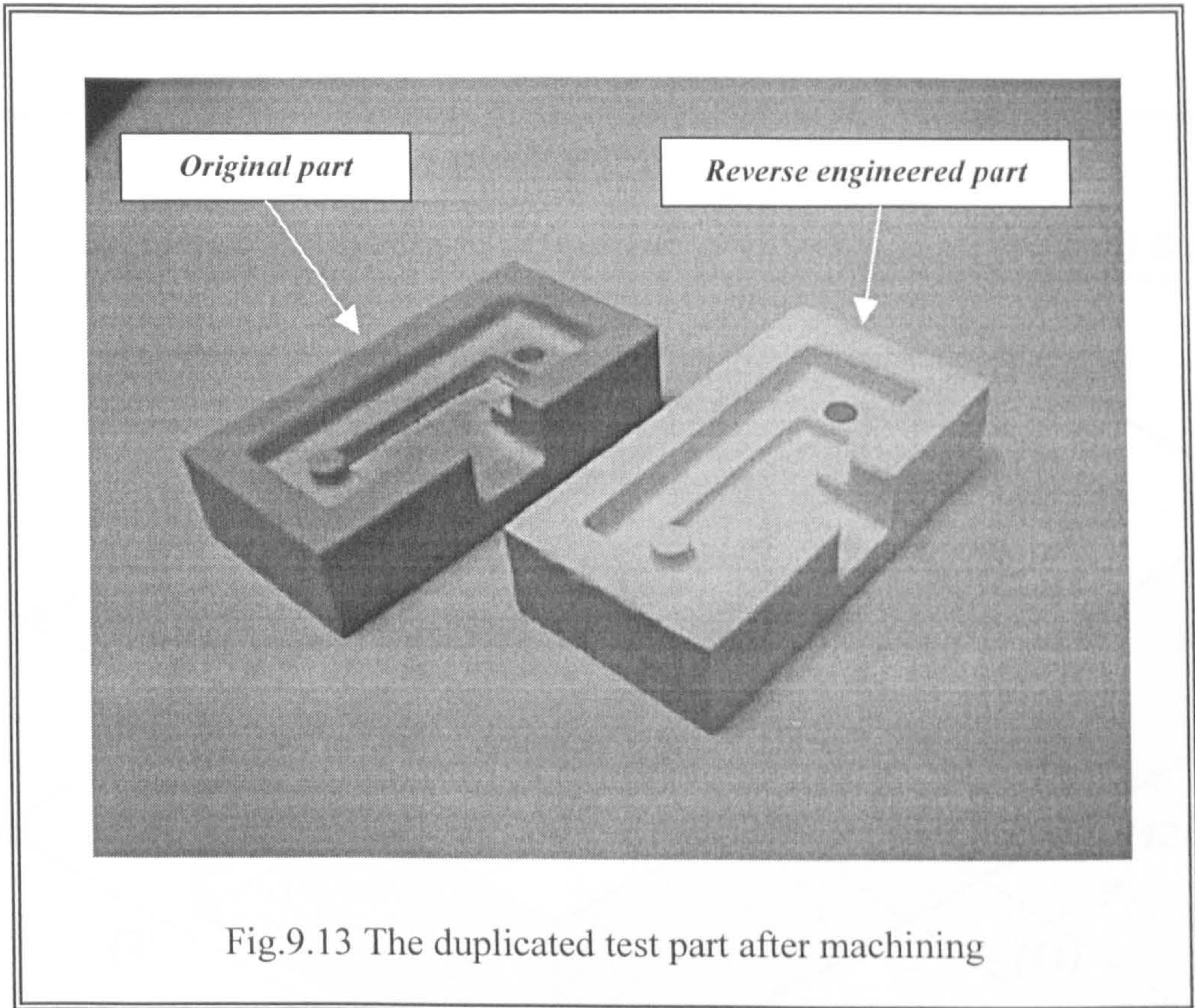


Fig.9.12 NC machining



The reverse engineered product of this RE attempt is shown in Fig. 9.13.



Finally, in order to compute the geometric differences between the original and reverse engineered models, the reverse engineered part was inspected using the *Cyclone*. To fully characterise the form errors, both the overall part dimensions and each feature were measured. A uniformly sampled set of reference points on two models was generated. *Surfacer<sup>TM</sup>* and *ReTool* system were used to find the distance to the closest surface point between two models as shown in Fig. 9.14. The reference points are: ① overall dimension (1, 2, 3); ② the centre and diameter of boss and hole features (4, 5, 6, 7); ③ the position and dimension of pocket (8, 9, 10); ④ the



position and dimension of slot (11, 12, 13); ⑤  $3 \times 5$  grid of points on free form surface (14, 15, . . . , 22).

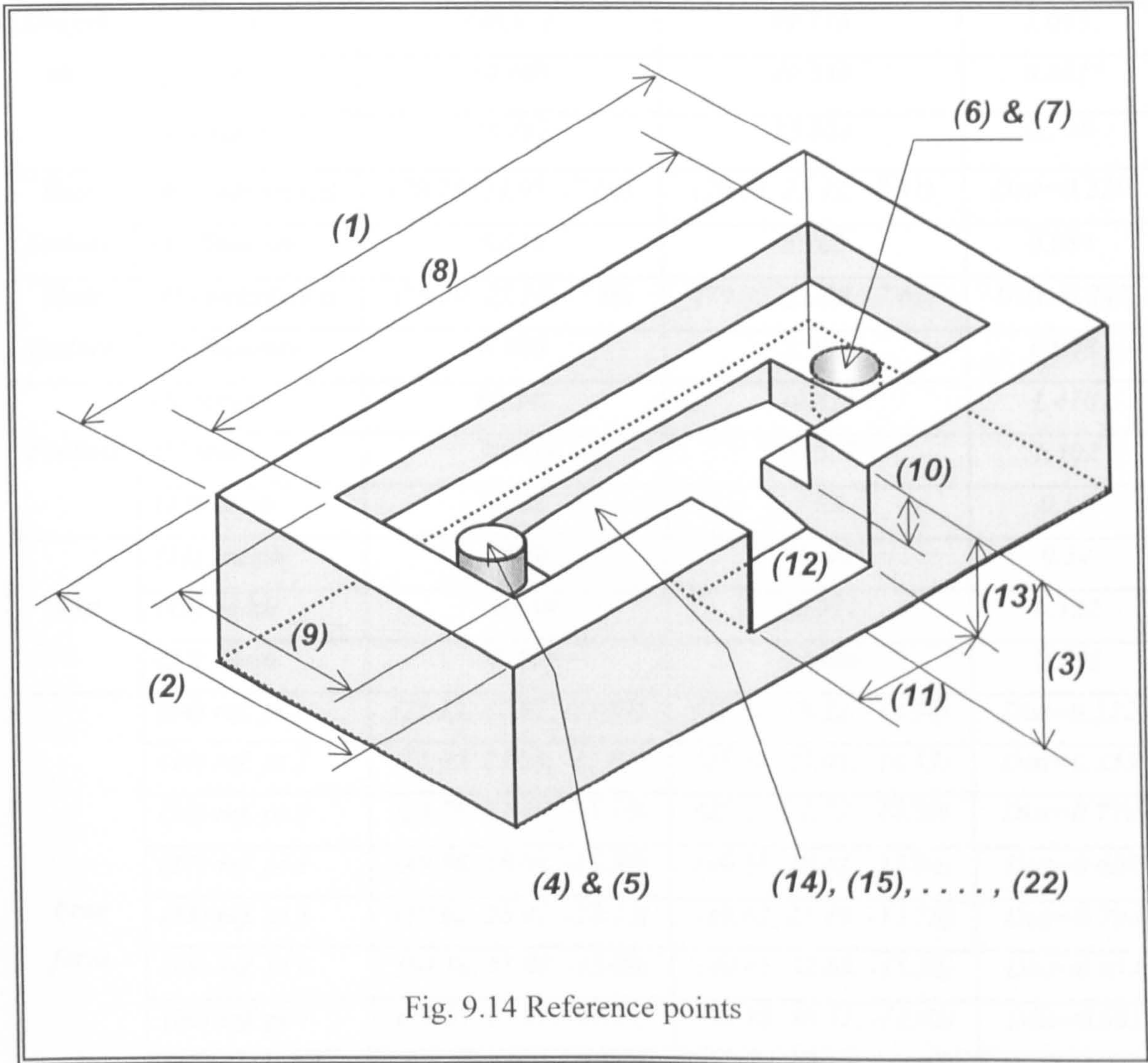


Fig. 9.14 Reference points

The deviation errors of all reference points between the typical dimension of two physical parts (original and reverse engineered) are compared in Table 9.1.

Table 9.1 Error analysis results (*Dist*: distance, *ref. pt*: reference point)

<i>Ref. points</i>		<i>Parts</i>	<i>Original (mm)</i>	<i>Reverse engineered (mm)</i>	<i>Error (mm)</i>
<i>Overall size</i>	(1) <i>length</i>		100.871	99.776	1.095
	(2) <i>width</i>		50.160	49.339	0.821
	(3) <i>height</i>		23.762	23.054	0.708
<i>Boss feature</i>	(4) <i>centre(x,y,z)</i>		(20.29, 24.95, -7.60)	(20.43, 25.12, -7.61)	<i>Dist</i> =0.220
	(5) <i>diameter</i>		8.019	8.103	0.084
<i>Hole feature</i>	(6) <i>centre(x,y,z)</i>		(79.99, 25.25, -7.58)	(79.82, 25.08, -7.62)	<i>Dist</i> =0.243
	(7) <i>diameter</i>		8.026	7.938	0.088
<i>Pocket1</i>	(8) <i>length</i>		80.341	79.811	1.470
	(9) <i>width</i>		29.859	28.557	1.302
	(10) <i>depth</i>		7.32	7.85	0.53
<i>Slot</i>	(11) <i>length</i>		18.70	18.36	0.34
	(12) <i>width</i>		19.689	20.211	0.522
	(13) <i>depth</i>		15.219	14.436	0.783
<i>Free form</i>	(14) <i>ref. pt 1</i>		(25.12, 17.97, -14.87)	(25.31, 18.22, -14.94)	<i>Dist</i> =0.322
	(15) <i>ref. pt 2</i>		(25.05, 25.08, -15.07)	(25.14, 25.03, -14.73)	<i>Dist</i> =0.355
	(16) <i>ref. pt 3</i>		(24.79, 32.16, -15.10)	(25.21, 32.77, -14.89)	<i>Dist</i> =0.770
	(17) <i>ref. pt 4</i>		(49.98, 18.04, -15.31)	(49.58, 18.41, -15.04)	<i>Dist</i> =0.608
	(18) <i>ref. pt 5</i>		(50.02, 25.01, -12.13)	(49.62, 25.19, -12.78)	<i>Dist</i> =0.784
	(19) <i>ref. pt 6</i>		(50.10, 31.96, -15.08)	(49.65, 32.63, -15.36)	<i>Dist</i> =0.854
	(20) <i>ref. pt 7</i>		(75.01, 17.99, -15.15)	(75.45, 18.51, -15.02)	<i>Dist</i> =0.693
	(21) <i>ref. pt 8</i>		(75.15, 25.12, -15.01)	(75.57, 25.06, -15.18)	<i>Dist</i> =0.471
	(22) <i>ref. pt 9</i>		(74.99, 32.10, -14.89)	(75.29, 32.43, -15.05)	<i>Dist</i> =0.473

Reference points (1), (2), and (3) denote the overall dimension of the part (*length*  $\times$  *width*  $\times$  *height*), respectively. From the results, the reverse engineered part is smaller than original, indicating size reduction. This is because of the limitations of tactile



scanners and its strategies since it is not always possible to scan the data points exactly on the boundaries of a face or feature. This results in a bounding area for the scanned data points to be smaller than the actual surface. Reference points (4), (5), (6), and (7) were selected to assess cylindrical features such as boss and hole. From the result of inspection test, the error of cylindrical features was founded to be the smallest value. This means that hole and boss feature can accurately be reconstructed. In order to test pocket and slot features, reference points from (8) to (13) were chosen. The pocket and slot features have a nominal dimensional error of approximately  $1.0\text{ mm}$  and  $0.5\text{ mm}$ , respectively. It can be seen that the variation in error decreased as the size of features decreased. The nominal error of free form surface was less than  $0.2\text{ mm}$  along to the z-direction.

The nature of the error is particularly crucial since the possibility of compensation can be searched. Errors were inevitably introduced and were cumulative. These errors are caused by a number of factors including scanning accuracy, modelling approximations, truncation errors, and manufacturing processes.



# Chapter 10

## Discussion and Conclusions

### 10.1 General Discussion and Conclusions

Reconstructing solid models directly from a scanned point set is of paramount importance since the resulting reconstructed CAD models can be directly imported into commercial solid modellers for process planning, integral property computation, assembly analysis and manufacturing.

The prototype of a novel RE system, ReTool, has been developed to implement the concept of feature-based reverse engineering. A key feature of the developed system is the incorporation of ANNs in feature recognition process. The use of ANN approach has enabled the developed system to be easily trained to handle new types of features. The ReTool system works with 2D inputs and is therefore more straightforward than the existing feature recognition algorithm that operates directly on 3D inputs. Features such as *block*, *pocket*, *step*, *slot*, *hole*, and *boss* can automatically be recognised from 2D geometric attributes – *chain codes*, *circular/rectangular*, *convex/concave*, and *open/closed* – that are directly derived

from 3D scanned points. Recognising each feature requires the determination of these four attributes. Those current features are very common and crucial in mechanical engineering products.

The user interface and the CAD model visualisation of the proposed system are implemented using Visual C++™ MFC and OpenGL™ in Windows systems. For the internal geometry representation, an application program interface (API) kernel is written and used. The ReTool system allows the representation of free form and prismatic features within one data structure using the B-rep solid modelling paradigm.

The ANN module is written in C-language. A number of experiments have been carried out to investigate the performance of the system to recognise features based on their shape. The test results show that the strategy for recognising features is reliable. It is because the network was trained sufficiently and properly. Moreover, the performance of recognition was excellent in terms of speed due to the nature of the ANNs.

The processing time is not discussed and compared because it mainly depends on the computer hardware.

The entire RE procedure to reconstruct CAD models implemented consists of the following:

1. Scanning of the part geometry by a contact or non-contact scanning device, as appropriate;
2. Applying data reduction module and detecting edge points to initial scanned points;
3. Segmentation;
4. Deriving geometric attributes from segmented point data;
5. Recognising a feature by inputting geometric attributes to ANN (if it is a free form surface, B-spline module is applied to fit a surface to segmented points);
6. Extracting the size and orientation of recognised features;
7. Transferring the reconstructed CAD models into other CAD/CAM systems using IGES neutral file format, if necessary;
8. The actual production of the reconstructed part;



## 10.2 Detailed Discussion and Conclusions according to Sub-Modules

- **Data acquisition and reduction**

A part surface is measured by using either contact scanners or non-contact scanners. Whilst all these systems produced 3D point data, the data vary widely in the point density and accuracy ranges they provide. The format of scanned point data by a contact device is usually uni-directional, but non-contact devices often generate unstructured point data. The selection of 3D scanner is dependent on the speed, accuracy and part shape. Detailed comparisons between contact and non-contact scanners were compared with their advantages and disadvantages. In short, contact scanners are usually superior to non-contact scanners in accuracy, whilst the speed of contact scanners comparing with non-contact scanners is a significant disadvantage.

Common scanning strategies were also examined and categorised into three types as follows. (a) a scanned data set with meshes, strips, and points, (b) a scanned data set with strips and points, (c) a scanned data set with points only.

- **Point data processing**

Generally the number of scanned points collected is likely to be more than necessary, so it is desirable to appropriately reduce the data volume by removing the redundant points.

There are two processes to refine a point cloud set implemented in this research: filtering and sampling.

The data filtering module is implemented using a moving average algorithm to remove the measurement noise of initial scanned points. This had the effect of smoothing out small deviations between scanned points, especially if the points are fairly close in value. After filtering, it has shown that the magnitude of curvature is reduced and the scanned point set is smoothed well. Since a high-accuracy contact scanner is used to scan a part, the uncertainty in different scanned points did not vary much. Therefore, for purposes of simplicity, the module for removing outliers is not considered in the proposed application.

For improving system memory and efficiency, two data sampling modules are developed and used to remove redundant scanned points; sampling by vector manipulations and sampling by rate. The sampling by vector manipulations selectively removes redundant points depending on the quality of the input point clouds, while retaining the information present in the data to a specified accuracy. This process is quite similar to the developed edge detection algorithm and very useful for sampling down a dense scanned point set.

The second method, sampling by rate, is simple but powerful technique to reduce highly redundant points within a given tolerance. It sampled every  $a^{\text{th}}$  point along scan lines, where  $a$  is an users-input integer value. That is to say, every point which sequence number is multiple of  $a$  are removed. Any scanned point satisfied by at least one of the above two sampling criteria is sampled.

- **Edge detection**

Edges are very crucial features for effective segmentation and object recognition. In this work, the dimensions of extracted features are also computed from detected edge points.

A reliable edge detection module for both sharp and smooth edges is proposed and implemented. The sharp edge detection technique is based on the computation of a local derivative operator. All the points at which tangency change value is greater than a *predefined tolerance* are detected as sharp edge points.

In order to detect smooth edge points, curvature differences between two consecutive neighbouring points are computed. Curvatures are computed using following two methods. One method is to compute curvatures by calculating a circle through three consecutive scanned points. The other method is to calculate curvatures by fitting a B-spline curve to a series of scanned point data. ‘*Spike*’ provides a reasonable solution for detecting smooth edges in a curvature plot. These spikes in the curvature plot give a clear indication of the location at which the point data are to be partitioned into different sets. Hence, the presence of spikes is used to detect the presence of smooth edges in the part.

From the results of various tests, the proposed edge detection modules are reliable.



- **Segmentation**

Segmentation is to manually divide the initial point data into subsequent regions according to their shapes, so that each region can be divided into a free form surface or a suitable distinctive prismatic feature. Fitting-surface or extracting-feature is carried out only on the given region (segment). In the prismatic shaped regions, therefore, the appropriate feature is extracted to give a more accurate model, whilst a B-spline surface is fitted to a free form surface region.

Segmentation is a vital step towards the interpretation of discrete scanned data, and the level of subdivision depend on the complexity of the part. The segmentation of input point data is accomplished manually. The mouse is used to select enough points by defining four corners on the displayed image using a 'window' function. With the help of this 'window' segmentation, whole reconstruction process regarding feature extraction is divided into a set of certain segmented area.

- **Free form surface creation by B-spline techniques**

Free form surfaces are commonly used in the design of new parts for many reasons including aerodynamic advantage and improved aesthetic. The most typical problem of surface creation is the difficulty in generating surface models with the low precision control and the problem associated with composite surfaces modelling. Among the parametric polynomial surface models such as *Ferguson surfaces*, *Bezier surfaces*, *B-spline surfaces* and *non-uniform rational B-spline surfaces (NURBS)*, B-spline technique is one of the most widely accepted

modelling functions in an interactive computer environment because of their ease of representation and applicability.

In this research, a module for creating free form surfaces from a 3D scanned point data set is designed and developed using B-spline modelling techniques. The B-spline module is presented which combines 3D scanned points and associated CAD surfaces for the reconstruction and modification of complex free form surfaces. Moreover, the extracted features can be transformed to B-rep solid models bounded by planes or B-spline surfaces.

The number of control points was adjusted to be the same to that of data points, although it required more storage space. Hence, the average deviation between the input point data and the reconstructed surface was almost *zero* units at the spot of input scanned points. These methods are based on the assumption that the points were arranged in a grid and are therefore not suitable for randomly scattered scanned points.

Using the IGES 128 format, the created surfaces are transferred to various major commercial CAD/CAM systems.

A trimmed surface is a surface that is essentially a subset of a supporting surface, and it is defined by a set of trimming curves that describe the boundary of that part of the whole surface that it occupies. They play a vital role in CAD/CAM process. Mathematical forms of trimmed surfaces are also discussed.

- **B-rep based models**

A B-rep solid is represented as a volume contained in a set of surfaces together with topological information which defines the relationships between the surfaces. In this work, the reconstructed CAD models such as features or surfaces were transformed to B-rep based models. Thus, B-rep based representation was significant to interface with commercial applications for a variety of engineering activities. Current work was focused on the validity of the algorithm with speed, robustness, and efficiency.

The data structures of four topological elements used in the application were *Body*, *Loop*, *Edge*, and *Vertex* structure. However, B-rep models can represent a wide range of objects but the data structure is complex, and it requires a large memory space.

- **Feature library**

The use of features as geometric primitives provides substantial advantages in usability and efficiency. There are various types of geometric features that are used in product definition. For each application, an appropriate feature library containing the generic definitions of all atomic features is required for performing the necessary tasks of the application. The number of features in the library reflects the range of the parts and applications that must be covered.

In this work, the STEP classification of volume feature taxonomy is used as the basis of the feature library. This proposed feature library is composed of six 2.5D



generic features such as *pocket*, *hole*, *boss*, *block*, *step*, and *slot*. The feature set can easily be expanded further by newly training the ANN module with appropriate training data. The shape of each generic feature was defined by size dimension and orientation on the part model.

- **Feature recognition using ANN**

There have already been several feature recognition methods so far: graph-based recognition, set-based recognition (volume decomposition), rule-based recognition and syntactic-based recognition. However, most research works for extracting features are intended to work with existing solid models using a graph relationship among vertex, edge, and face.

The feature recognition approach presented in this research is unique, since an ANN is employed with its learning and parallel processing capabilities. The ANN module has been written in C-language.

A general three-layer neural network is employed. The present ANN consists of an input layer with 10 neurons, a hidden layer of 15 neurons, and an output layer of 6 neurons. The back-propagation learning involved with  $\eta$  (learning rate) = 0.9,  $\mu$  (momentum term) = 0.5,  $X_0$  (bias term) = 1.0, and  $\varepsilon$  (error tolerance) = 0.001. With these parameters and all training patterns, the ANN algorithm is trained completely after all the training data is presented to update the weights of connections with around 70 000 times without any error.

ANNs require parallel input patterns. The input patterns are typical geometric attributes of an input scanned point set that classify features such as *chain codes*, *convex/concave*, *circular/rectangular*, and *open/closed attribute*. These input attributes are automatically derived from the selected scanned points by the developed robust modules. The derivation of these input attributes is based on a set of general vector manipulations with selected *3D* scanned points. Each input pattern was defined on arbitrary *2D* plane. Hence, it was seen that *3D* geometric information can be produced by combining some of different *2D* information. Then, they were presented to the ANN module to classify an appropriate feature.

The output of ANN module is the type of one of the features in the feature library according to the output value of the neuron that was high, whilst all the others were low. When more than one output is high, or none of the outputs are high, the pattern to the class of the output neuron with the highest numerical value was simply assigned.

The number of features in the feature library can be further expanded depending on the characteristics of applications. Although this research presented example features with seven chain codes, it can certainly handle features with more chain codes. This means that if the number of chain codes is increased, the shape of recognised feature can be more complex. However, an increased number of chain codes requires neural network with more neurons in the input layer, output layer, and possibly in hidden layer. This neural network needs more time during training process.

The capability of feature recognition using the proposed strategy was satisfactory. Generic features were well extracted from input scanned data to CAD environment.

- **Feature extraction**

Once CAD models are recognised from a selected scanned point data set, then their location, orientation, and dimension information needs to be extracted.

In particular, in order to extract the exact size and location of circular shaped features, the centre position and radius of a circle were estimated directly from scanned points using an iteration LMS method that minimises the radius and the centre estimation error. This algorithm converged to the true circle reasonably quickly with the best estimated circle centre and its radius.

A reconstruction function needs to be chosen to measure the agreement between the scanned data and the reconstructed *CAD* models with a particular choice of parameter. In this work, a MLE theory was adopted and used, which presents the likely errors of the best-fit parameters. The objectiveness of this statistical method is to estimate the parameters of the reconstruction function that most likely correspond to the previously detected edge points.

It is not straightforward to support design change of a sophisticated model. Even when there exist a similar CAD model, a new model is often re-designed without any support of the existing model. From that point of view, the proposed system introduces interactivity after the extraction assuming that the result of the extraction is only a first approximation and the user must edit the final result. This gave more flexibility to the system in context of *parametric design*. Parametric design scheme



is helpful to modify the existing CAD model to a new model. That is, the user can modify the geometry by varying the feature parameters.

- **CAD model exchange using IGES**

A CAD system usually creates CAD models by its own internal representation. Hence, there is a need to share product data between the various applications used during design, analysis, process planning, and manufacturing processes for reducing the product development cost. Thus, a standard file format is necessary to convey varied geometric descriptions among numerous CAD/CAM/CAE systems. IGES is a standard used to exchange product data between different commercial CAD/CAM/CAE systems.

In this research, the reconstructed CAD models stored in the proposed database of ReTool system are compatible with IGES data file format, so that output files are exchangeable with other applications for further engineering activities. Using this approach to the data format gave more flexibility to different commercial applications.

### **10.3 Limitation of the Proposed Application**

The limitations of the proposed method are:

- (a) the input scanned point clouds have to be structured data;
- (b) the part is composed of prismatic features that can be defined beforehand and no excessively entangled compound feature exists in the part;

## **10.4 Further Research**

It is not the target of the proposed application to compete with commercial CAD systems. It is more crucial to concentrate on the development of solutions to specific problems within the RE process. With such a premise, the proposed application can be used as an 'add-on' to or be integrated into an existing CAD package.

Further work needs to be done to make the feature library even more general in order to deal with secondary features such as chamfers, fillets, rounds, and taps. Each of these involves small scale geometry that requires specialised gauges for accurate measurement. A user accessible feature definition system should also be developed, so that the feature database can be modified and extended by administrators according to specific needs.

Due to limitations of tactile scanners and its strategies, it is not always possible to scan the data points exactly on the boundaries of a face or feature. This results in a bounding area for the scanned data points to be smaller than the actual surface. Hence, an algorithm needs to be developed to compensate the gap.



## References

- Abella, R.J., Daschbach, J.M. and McNichols, R.J. (1994) Reverse engineering industrial applications, *Computers Industrial Engineering*, 26(2), pp 381–385.
- Allada, V. and Anand, S. (1995) Feature-based modelling approaches for integrated manufacturing: state-of-the-art survey and future research directions, *International Journal of Computer Integrated Manufacturing*, 8(6), pp 411–440.
- Alschuler, D.M. (1988) Parametric Design with Solids, *Computer-Aided Engineering*, Sept, pp 36–40.
- Ballard, D.H. (1996) Generalising the Hough transform to detect arbitrary shapes, *Pattern Recognition*, 13, pp 111–119.
- Bauert, F. *et al* (1990) Modelling methods for a flexible computer aided embodiment design system, *Research in Engineering Design*, 2, pp 15–34.
- Baumgart, B.G. (1972) Winged-edge polyhedron Representations, Technical Report STAN-CS-320, Computer Science Department, Stanford University.
- Besl, P.J. and McKay, N.D. (1992) A method for registration of 3-D shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), pp 239–255.

- Bidanda, B. and Hosni, Y.A. (1994) Reverse engineering and its relevance to industrial, *Computers Industrial Engineering*, 26(2), pp 343–348.
- Blanc, C. and Schlick, C. (1996) Accurate parameterisation of conics by NURBS, *IEEE Computer Graphics and Applications*, November, pp 64–71.
- Bolle, R.M. and Cooper, D.B. (1986) On optimally combining pieces of information, with application to estimating 3D complex-object position from range data, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(5), pp 619–638.
- Bradley, C. and Vickers, G.W. (1993) Free - form surface reconstruction for machine vision rapid prototyping, *Optical Engineering*, 32(9), pp 2191–2200.
- Bronsvoort, W.F. and Jansen, F.W. (1993) Feature modelling and conversion-key concepts to concurrent engineering, *Computers in Industry*, 21, pp. 61–86.
- Brown, C.W. (1990) An Automated Planning System for Dimensional Inspection, 22<sup>nd</sup> CIRP International Seminar on Manufacturing Systems, June 11<sup>st</sup>–12<sup>nd</sup>.
- Buurman, J. and Bierhuizen, D.J.W. (1990) A two stage object identification system in the deft intelligent assembly cell, *Machine Vision Systems Integration in Industry*, SPIE 1386, pp 185–196.
- Chang, K.H. (1998) Reverse Engineering for detailed Inspection and Data Analysis, MSc thesis, The University of Warwick, UK.

- Chen, Y.H. and Liu, C.Y. (1997a) Robust segmentation of CMM data based on NURBS, *International Journal of Advanced Manufacturing Technology*, 13, pp 530–534.
- Chen, L.C. and Lin, G.C.I. (1997b) A vision-aided reverse engineering approach to reconstructing free-form surfaces, *Robotics and Computer-Integrated Manufacturing*, 13(4), pp 323–336.
- Chen, Y.H. and Lee, H.M. (1998) A neural network system for two-dimensional feature recognition, *International Journal of Computer Integrated Manufacturing*, 11(2), pp 111–117.
- Chen, Y.H. and Liu, C.Y. (1999) Quadric surface extraction using genetic algorithms, *Computer-Aided Design*, 31, pp 101–110.
- Chivate, P.N. and Jablokow, A.G. (1993) Solid-model generation from measured point data, *Computer-Aided Design*, 25(9), pp 587–599.
- Chivate, P.N., Puntambekar, N.V. and Jablokow, A.G. (1999) Extending surfaces for reverse engineering solid model generation, *Computers in Industry*, 38, pp. 285–294.
- Choi, B., Barash, M. and Anderson, D. (1984) Automatic recognition of machined surfaces from a 3D solid models, *Computer-Aided Design*, 16(2), pp 81–86.
- Chua, C.K. *et al* (1998) An integrated Experimental approach to link a laser digitiser, a CAD/CAM system and Rapid Prototyping system for biomedical applications, *International Journal of Advanced Manufacturing Technology*, 14, pp 110–115.



- Devireddy, C.R. and Ghosh, K. (1999) Feature-based modelling and neural networks-based CAPP for integrated manufacturing, *International Journal of Computer Integrated Manufacturing*, 12(1), pp 61–74.
- Donaldson, I.A. and Corney, J.R. (1993) Ruled-based feature recognition for 2.5D machined components, *International Journal of Computer Integrated Manufacturing*, 6, pp 51–64.
- Duncan, J.P. and Mair, S.G. (1983), *Sculptured surfaces in engineering and medicine*, Cambridge University Press.
- Editorial (1997) Special Issue: reverse engineering of geometric models, *Computer - Aided Design*, 29(4), pp 253–54.
- Faux, I.D. and Pratt, M.J. (1981) *Computational Geometry for Design and Manufacture*, Ellis Horwood Ltd.
- Freeman, A. and Skapura, D.M. (1992) *Neural Networks; Algorithms, Applications, and Programming Techniques*, Addison-Wesley Publisher.
- Galt, A. (1997) *Contact Scanning, there are no boundaries*, Renishaw plc, April.
- Gindy, N.N.Z (1989) A hierarchical structure for form features, *Int. J. Prod. Res.*, 27(12), pp 2089–2103.
- Goh, A., Hui, S.C., Song, B. and Wang, F.Y. (1997) A STEP/EXPRESS to object-oriented database translator, *International Journal of Computer Applied Technology*, 10(1/2), pp 90–96.
- Gonzalez, R.C. and Woods, R.E. (1992) *Digital image processing*, Addison-Wesley Publishing Co.
- Grabouski, H. (1990) Dialogue oriented intelligent CAD system, *Journal of Engineering Design*, 1, pp 33–52.

- Green, P.A. and Philpott, M.L. (1994) Error modelling of Reverse Engineered free form surfaces, *Transactions of NAMRI/SME*, XXII, pp 259–266.
- Gu, P. and Norrie, D.H. (1995a) *Intelligent Manufacturing Planning*, Chapman & Hall, 2-6 Boundary Row, London.
- Gu, P. and Yan, X. (1995b) Neural network approach to the reconstruction of freeform surfaces for reverse engineering, *Computer-Aided Design*, 27(1), pp 59–64.
- Hafele, K.H. (1996) POMOS – Point based Modelling System, B G Teubner Stuttgart, pp. 63–81.
- Hoffmann, C.M. (1989) *Geometric and Solid Modeling*, Morgan Kaufmann Publisher, Inc. San Mateo, California 94403.
- Hosni, Y.A. and Ferreira, L. (1994) Laser based system for reverse engineering, *Computers Industrial Engineering*, 26(2), pp 387–394.
- Hu, Y.P. and Sun, T.C. (1997) Moving a B-spline surface to a curve – a trimmed surface matching algorithm, *Computer-Aided Design*, 29, pp 449–455.
- Huang, C.N. and Motavalli, S. (1994) Reverse engineering of planar parts using machine vision, *Computers Industrial Engineering*, 26(2), pp 369–379.
- Jablokow, A.G. and Abraham, I. (1993) Reconstruction of B-rep solid models from finite element meshes for design automation, DE-Vol. 65-2, *Advances in Design Automation – Volume 2*, ASME, pp 47–59.
- Jain, P.K. and Kumar, S. (1998) Automatic feature extraction in PRIZCAPP, *International Journal of Computer Integrated Manufacturing*, 11(6), pp 500–512.

- Joshi, S. and Chang, T.C. (1988) Graph-based heuristics for recognition of mechanical features from a 3D solid model, *Computer-Aided Design*, 20(2), pp 58–66.
- Jun, Y.T., Lee, S.J., Choi, J.B. and Park, S.H. (1993) Generation of trimmed surfaces in a surface modelling system, *Journal of the Korean Society of Mechanical Engineers*, 17(6), pp. 1372–1380.
- Kim, Y.S. (1992) Recognition of form features using convex decomposition, *Computer-Aided Design*, 24(9), pp 461–476.
- Kohonen, T. (1988) An introduction to neural computing, *Neural Networks*, 1, pp 3–16.
- Lai, J.Y. and Lu, C.Y. (1996) Reverse Engineering of composite sculptured surfaces, *International Journal of Advanced Manufacturing Technology*, 12, pp 180–189.
- Landau, U.M. (1987) Estimation of a circular arc centre and its radius, *Computer Vision, Graphics, and Image Processing*, 38, pp 317–326.
- Lazar, M. and Wagner, T. (1995) Reverse Engineering in Rapid Prototyping, *Proceedings of the 4TH European conference on RP&M*, June, pp 105–122.
- Lee, E.T.Y. (1989) Choosing nodes in parametric curve interpolation, *Computer-Aided Design*, 21(6), pp 363–370.
- Lin, C.Y., Hwang, Y.Y. and Lai, J.Y. (1997) Interference-free cutting-path generation based on scanning data, *International Journal of Advanced Manufacturing Technology*, 13, pp 535–547.



- Lin, A.C. and Chen, C.F. (1998) Point-data processing and error analysis in Reverse Engineering, *International Journal of Advanced Manufacturing Technology*, 14, pp 824–834.
- Ma, W. and Kruth, J.P. (1995) Parameterisation of randomly measured points for least squares fitting of B-spline curves and surfaces, *Computer-Aided Design*, 27(9), pp 663–675.
- Mantyla, M. (1985) *An Introduction to Solid Modeling*, Computer Science Press, Rockville, MD.
- Medoni, J. and Yasumoto, Y. (1987) Corner detection and curve representation using cubic B-splines, *Computer Vision, Graphics and Image Processing*, 39, pp 267–278.
- Menq, C. and Chen, F.L. (1996) Curve and surface approximation from CMM measurement data, *Computers Industrial Engineering*, 30(2), pp 211–225.
- Merat, F.L. and Radack, G.M. (1992) Automatic inspection planning within a feature-based CAD system, *Robotics & Computer-Integrated Manufacturing*, 9(1), pp 61–69.
- Milroy, M.J., Weir, D.J., Bradley, C. and Vickers, G.W. (1996) Reverse Engineering employing a 3D laser scanner: A case study, *International Journal of Advanced Manufacturing Technology*, 12, pp 111–121.
- Milroy, M.J., Bradley, C. and Vickers, G.W. (1997) Segmentation of a wrap-around model using an active contour, *Computer-Aided Design*, 29(4), pp 299–320.
- Mortenson, M.E. (1985) *Geometric Modeling*, New York: John Wiley.

- Motavalli, S. and Shamsaasef, R. (1996) Object-oriented modelling of a feature-based reverse engineering system, *International Journal of Computer Integrated Manufacturing*, 9(5), pp 354–368.
- Motavalli, S. and Valenzuela, J. (1998) A system for reverse engineering of prismatic parts using orthographic images, *International Journal of Computer Integrated Manufacturing*, 11(2), pp 103–110.
- Mukherjee, S. and Gurumoorthy, B. (1994) Surface approximation in Reverse Engineering, *Advances in Design Automation*, Volume 1, ASME, pp 333–336.
- Nezis, K. and Vosniakos, G. (1997) Recognising 2.5D shape features using a neural network and heuristics, *Computer-Aided Design*, 29(7), pp 523–539.
- ParaSolid™ (1998) Online Manual
- Park, S. and Park, M. (1997) Development of a software for 3D surface modelling and optimum machining, Technical Report, Korea Institute of Science of Technology (in Korean).
- Piegl, L. and Tiller, W. (1995) *The NURBS Book*, Springer-Verlag, New York, NY.
- Podehl, G. (1996) Quo vadis Reverse engineering?, B. G. Teubner Stuttgart, Germany, pp 169–171.
- Pratt, M.J. (1990) Synthesis of an optimal approach to form feature modelling, *Computer Graphics*, 14, pp 225–235.
- Prebhakar, S. and Henderson, M.R. (1992) Automatic form-feature recognition using neural-network-based techniques on boundary representations of solid models, *Computer-Aided Design*, 24(7), pp 381–393.

- Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P. (1992) *Numerical Recipes in C*, Cambridge University Press.
- Puntambekar, N.V., Jablakov, A.G. and Sommer III, H.J. (1994) Unified review of 3D model generation for reverse engineering, *Computer Integrated Manufacturing Systems*, 7(4), pp 259–268.
- Raja, V.H. and Lawrie, C.C. (1997) The scope of Reverse Engineering within Engineering Businesses, *Engineering Development International*, Official Publication of FEANI.
- Rekoff, M.G.JR. (1985) On reverse engineering, *IEEE Trans. On System, Man and Cybernetics*, SMC-15, pp 244–252.
- Renishaw (1998) Brochure on Scanning System.
- Sarraga, S. and Waters, W. (1984) Free-form surfaces in GMSOLID: Goals and Issues, *Proceedings of a symposium on Solid Modelling by Computers*, pp 187–209.
- Sarkar, B. and Menq, C.H. (1991) Smooth - surface approximation and Reverse Engineering, *Computer - Aided Design*, 23(9), pp 623–628.
- Seiler, A., Balendran, V., Sivayoganathan, K. and Sackfield, A. (1996) Reverse Engineering from uni-directional CMM scan data, *International Journal of Advanced Manufacturing Technology*, 11, pp 276–284.
- Shah, J.J. (1990) The ASU features test bed: an overview, *Computer in Engineering*, 1, pp 233–241.
- Shah, J.J. (1992) Features in Design and Manufacturing, Chapter 2, *Intelligent Design and Manufacturing*, John Wiley & Sons, Inc., pp 39–71.



- Shah, J.J. and Mantyla, M. (1995) *Parametric and Feature-based CAD/CAM (Concepts, Techniques, and Applications)*, John Wiley & Sons, Inc.
- Sheng, Y. and Chevrette, P.C. (1998) Three-dimensional object recognition from two-dimensional images using wavelet transform and neural networks, *Optical Engineering*, 37(3), pp 763–770, March.
- Shum, S.S.P., Lau, W.S., Yuen M.M.F. and Yu, K.M. (1997) Solid reconstruction from orthographic opaque views using incremental extrusion, *Computer & Graphics*, 21(6), pp 787–800.
- Sinha, S. and Seneviratne, P. (1996) *Reverse Engineering*, B G Teubner Stuttgart, pp 7–18,.
- Stephen, P.B. and Adrian, B (1993) Feature Technology, Technical Report 001/1993, School of Mechanical Engineering, University of Bath, Bath, BA2 7AY, UK.
- Theodoracatos, V.E. and Bobba, V. (1993) NURBS surface reconstruction from a large set of image and world data points, *Advances in Design Automation*, ASME, 2, pp 15–33.
- Thomas, S.M. and Chan, Y.T. (1989) A simple approach for the estimation of circular arc centre and its radius, *Computer Vision, Graphics, and Image Processing*, 45, pp 362–370.
- Thompson, W.B., Owen, J.C. and James, H. (1995) Feature-Based Reverse Engineering of Mechanical Parts, University of Utah, Department of Computer Science, Technical Report UUCS-95-010, November 6.

- Tonshoff, H.K., Aurich, J.C. and D'Agostino, N. (1996) A unified approach to free-form and regular feature modelling, *Annals of the CIRP*, 14(1), pp 125–128.
- Toriya, H. and Chiyokura, H. (1993) *3D CAD (Principles and Applications)*, Springer-Verlag Berlin Heidelberg, Germany.
- Triform (1998) Brochure on Triform 3D Image Capture System, Wicks and Wilson Limited, Morse Road, Basingstoke Hampshire, England.
- Tyan, L.W. and Devarajan, V. (1998) Automatic identification of non-intersecting machining features from 2D CAD input, *Computer-Aided Design*, 30(5), pp 357–366.
- Ueng, W.D. and Lai, J.Y. (1998) A sweep surface fitting algorithm for reverse engineering, *Computers in Industry*, 35, pp 261–273.
- Varady, T., Martin, R.R. and Cox, J. (1997) Reverse engineering of geometric models – an introduction, *Computer-Aided Design*, 29(4), pp 255–268.
- Wang, M.T. (1991) An object oriented feature-based CAD-CAPP-CAM integration frame work, *Advances in Design Automation*, 10, pp 12–18.
- Wang, D. (1993) Pattern recognition: neural networks in perspective, *IEEE Expert*, August, pp 52–60.
- Warthen, B.D. (1992) What is STEP for?, *Product Data International*, 3(6), pp 1–10.
- Weiler, K. (1988) The radial Edge Structure: A Topological Representation for Non-Manifold Geometric Boundary Modelling, *Geometric Modelling for CAD Applications*, Elsevier Science Publishers.

- Werner, W. (1993) Digitisation – An alternative to copy milling, *European Product Engineering*, 17(3), pp 27–28.
- Woo, T.C. (1982) Feature extraction by volume decomposition, *Proc. Conf. On CAD/CAM technology in Mechanical Engineering*, Cambridge Mass., USA, pp 76–94.
- Woo, T.C. (1985) A combinational analysis of boundary data structure schemata, *IEEE Computer Graphics and Applications*, 5(3), pp 19–27.
- Wu, M.C., Chen, J.R. and Jen, S.R. (1994) Global shape information modelling and classification of 2D workpieces, *International Journal of Computer Integrated Manufacturing*, 7(5), pp 261–275.
- Wu, P.S.Y. and Yingen, X. (1997) A learning mechanism for parts recognition in an intelligent assembly system, *International Journal of Advanced Manufacturing Technology*, 13, pp 413–418.
- Yau, H.T. (1997) Reverse engineering of engine intake ports by digitisation and surface approximation, *International Journal of Machine Tools Manufacturing*, 37(6), pp 855–871.
- Zeid, I. (1991) *CAD/CAM Theory and Practice*, McGraw-Hill, Inc.
- Zhang, S.G. and Ajimal, A. (1995) Reverse Engineering and Its application in Rapid Prototyping and Computer Integrated manufacturing, *Computer Application in Production Engineering, Proceedings of CAPE'95*, Chapman & Hall, pp171–178.
- Zhang, C., Chan, K.W. and Chen, Y.H. (1997) A method for recognising feature interactions and feature components within the interactions, *International Journal of Advanced Manufacturing Technology*, 13, pp 713–722.



# Bibliography

- Faux, I.D. and Pratt, M.J. (1981) *Computational Geometry for Design and Manufacture*, Ellis Horwood Ltd.
- Freeman, J.A. and Skapura, D.M. (1992) *Neural Networks; Algorithms, Applications, and Programming Techniques*, Addison-Wesley Publisher Co.
- Gonzalez, R.C. and Woods, R.E. (1992) *Digital image processing*, Addison-Wesley Publishing Co.
- Gu, P. and Yan, X. (1995) Neural network approach to the reconstruction of freeform surfaces for reverse engineering, *Computer-Aided Design*, 27(1), pp 59–64.
- Hoffmann, C.M. (1989) *Geometric and Solid Modeling*, Morgan Kaufmann Publisher, Inc. San Mateo, California 94403.
- Piegl, L and Tiller, W. (1995) *The NURBS Book*, Springer-Verlag.
- Shah, J.J. and Mantyla, M. (1995) *Parametric and Feature-based CAD/CAM (Concepts, Techniques, and Applications)*, John Wiley & Sons, Inc.
- Toriya, H. and Chiyokura, H. (1991) *3D CAD: Principles and Applications*, Springer-Verlag.

Wu, P.S.Y. and Yingen, X. (1997) A learning mechanism for parts recognition in an intelligent assembly system, *International Journal of Advanced Manufacturing Technology*, 13, pp 413–418.

Zeid, I. (1991) *CAD/CAM Theory and Practice*, McGraw-Hill, Inc.

# Appendix 1

## A Guide to 3D Digitisers

<i>COMPANY</i>	<i>PRODUCT</i>	<i>APP. PRICE</i>	<i>TECHNOLOGY</i>	<i>WORK VOLUME</i>	<i>ACCURACY (INCHES)</i>	<i>SPEED</i>
<i>Cheasapeake Kennett Square, PA 610-444-2300</i>	<i>CMS-3000</i>	<i>\$150,000</i>	<i>Portable CMM Tracking Laser Interferometer</i>	<i>100'x100'x100'</i>	<i>0.0005</i>	<i>1000 points/sec Scanning</i>
<i>Cyberware Monerey, CA 408-657-1450</i>	<i>3030 RGB Hirez/MM</i>	<i>\$75,200</i>	<i>Line Laser</i>	<i>Varies</i>	<i>0.1 mm radially</i>	<i>14,556 points/sec</i>
<i>Digibotics Austin, TX</i>	<i>Digibot II</i>	<i>\$54,450</i>	<i>4-Axis point laser</i>	<i>18" dia.x18" cyl.</i>	<i>0.002</i>	<i>20 points/sec</i>



512-832-6544							
<i>EDIS</i> Santa Monica, CA 310-451-8566	<i>Mini-Moire</i>	\$39,000	Moire interferometry	0.5"x0.5" to 12"x12"	0.002-0.004	50,000 points/sec	
<i>Faro</i> <i>Technologies</i> Lake Mary, FL 407-333-9911	<i>FaroArm-B06-01/02</i> <i>FaroArm-B10-01/02</i> <i>FaroArm-S12-01/02</i>	\$14,400 \$24,900 \$76,400	6" Axis mechanical 6" Axis mechanical 6" Axis mechanical	6' Sphere 10' Sphere 12' Sphere	+ 0.012 + 0.016 + 0.007	Manual Manual Manual	
<i>Immersion</i> Corp. Palo Alto, CA 408-653-1160	<i>Immersion</i> <i>Personal</i> <i>Digitizer</i>	\$1495	Mechanical Arm	36" Sphere	0.03"	1000 points/sec	
<i>Hymarc</i> Ottawa, Canada 613-727-1584	<i>Hyscan</i>	\$75,000	"Flying Spot" laser scanning CMM/CNC retrofit	Equal to volume of CMM/CNC	0.0005"	10,000 points/sec continuous	
<i>Laser Design</i> Minneapolis, MN 612-884-9648	<i>Surveyor Series</i>	\$74,400- \$514,000	Single point triangulation laser Line Scan: triangulation scan	12"x6"x6" - 5"x 8"x 4"	Single Point: 0.0008 Line Scanner: 0.0001	Single point: 130 Line Scan: 14,400	
<i>Lemnie</i> Walled Lake, MI	<i>Lemoine-LDS</i>	\$59,400 base price	Mechanical Probe	Variable	0.0009	up to 20ipm	

810-960-5989 <i>Medar</i> <i>Farmington Hills, MI</i> 810-477-3900	<i>Cadeyes</i>	\$80,000	Moire interferometry	Varies from 5mm by 0.3mm to 300mm by 300 mm by 300	Small as 1 micron	25,000 points/sec
<i>Northern Digital</i> <i>Ontario, Canada</i> 519-884-5142	<i>Optotrak 3020</i>  <i>OptotrakRH-2020</i>	\$57,500  \$152,500	Infrared LED Linear CCD Infrared LED Linear CCD	Up to 2.60m by 3.5m at 6m from sensor Up to 4,15m by 4.9m at 8m from sensor	0.008  0.008	Manual  Manual
<i>Perceptron</i> <i>Farmington Hills, MI</i> 810-478-7710	<i>Laser</i>  <i>Tricam - C800</i>	\$50,000  \$13,500	Laser Radar  Laser Triangulation	45x45 degrees  79mmx167mm	2mm @ 1.75m Ambiguity Interval 0.05mm	100,000 to 390,000 points/sec 15,000 points/sec
<i>Pixsys</i> <i>Boulder, CO</i> 303-447-0248	<i>Flashpoint 3000</i>	\$18,000	Infrared LED/CCD	6.3' Cube	0.020	Manual
<i>Polhemus</i> <i>Colchester, VT</i> 802-655-3195	<i>Eastrak</i> <i>3Draw</i> <i>Isotrak II</i>	\$6350 \$6500 \$3175	Electromagnetic Electromagnetic Electromagnetic	30"x30"x30" 12"x18"x12" 30"x30"x30"	0.03" 0.010 0.1	120 updates/sec 120 updates/sec

							60 updates/sec
Renishaw Schaumburg, IL 708-843-3666	<i>Cyclone</i> <i>Retro Scan</i> <i>Renscan</i>	\$50,000 \$20,000 \$25,000	3-Axis Mechanical Probe SP2-1 Scanning Probe SP2-1 Scanning Probe	24"x20"x16"	0.0005	1000 points/sec 4.6 points/sec 68 points/sec	
SAC Shelton, CT 203-925-1661	<i>FreePoint 30</i>	\$5965	Sonic Triangulation	8' Cube	0.050	Manual	
Sharna Plymouth, MI 313-454-7192	<i>xtrolaser</i>	\$40,000	Single Point Laser (CMM/CNC)	Varies	0.001	140 points/sec	
Supraporte Romer Dearborn, MI 313-563-5933	<i>System 6-2000</i>	\$105,000	6-Axis handheld contact probe	6' Sphere & 8' Sphere	0.003-0.005	Manual	
3D Technology Trumbull, CT 203-371-8500	<i>Optical LS 25</i> <i>Replica 1000</i> <i>Persona 24</i>	\$48,000 \$75,000 \$135,000	Laser line triangulation Laser line triangulation Laser line triangulation	CNC/CMM machine dependent 40"x40"x18" 75"x40"	0.0008 0.002 0.05	Up to 14,000 (pts/sec) Up to 14,000 (pts/sec) 1000 (pts/sec)	

(Reproduced from: Terry Wohlers, "3D Digitizing for Engineering", *Computer Graphics World*, March 1995, pp. S2)



## Appendix 2

# Commercial RE Packages and Services Providers

### (a) Commercial Reverse Engineering Packages

<i>Company</i>	<i>Product Description</i>
<p><i>3D Scanners Ltd</i>  <i>London, UK</i>  <i>tel. + 44 171 922 8822</i>  <i>fax + 44 171 922 8899</i></p>	<p><i>RI SCAN</i> is used for controlling the scanning process and viewing the resultant data. <i>RI SCAN Pitch and Yaw</i> is an option in the <i>RI SCAN</i> software to increase the users capability to scan complicated objects.</p>
	<p><i>RI CAM</i> is used to generate NC toolpaths from scan data.</p>
	<p><i>RI MAKE</i> is used to manipulate scan data and to create 3D scan files from 2D bitmap data.</p>
	<p><i>RI INTERFACE</i> is used to prepare scan data for transfer to a 3D CAD/CAM system.</p>
	<p><i>RI TOOLS</i> is used to edit, filter and manipulate scan data.</p>

<p><i>Cyberware</i>  <i>Monerey, CA</i>  <i>408-657-1450</i></p>	<p><i>Echo</i> Cyberware's Echo software gives you interactive control of the entire digitising process. Once the digitising is completed to your satisfaction, you can use the Echo surface-editing tools to modify the model. Editing operations include scaling, cut and paste, smoothing, clipping, erasing, proportioning, mirror imaging, interpolating, and blending.</p>
	<p><i>Cymage</i> Cymage removes non-essential vertices from a Cyberware model by reducing the number of polygons by as much as 98%. This package will produce very efficient models in a wide variety of industry-standard file formats.</p>
	<p><i>CySculpt</i> CySculpt allows the user to edit and reshape digitised models. Editing and sculpting can be applied to individual vertices and facets, to free-form user-defined areas, and to the entire model. Operations include selective smoothing, roughening, stretching, decimation, and refinement.</p>
	<p><i>CyDir</i> CyDir Digitising Environment is a Motif-based interface for scanning objects, assembling and editing multiple scans, and merging multiple polygonal files into a single model.</p>
	<p><i>CyPie and CyEat</i> Cyberware introduces a technique in the assembly of multiple scans. Any number of scans can automatically be assembled into a complete view of a model.</p>
<p><i>Colo. St. Univ.</i></p>	<p><i>CAS-Computer Aided Sculpting</i> Allows interactive computer-assisted editing and free form sculpting of three-dimensional (3D) polygonal mesh surface models. CAS provides an excellent tool for editing digitised data</p>

	and provides an excellent interface between 3D digitising and rapid prototyping systems.
<i>Design Automation, Inc.</i> <i>3211-D Bramer Drive</i> <i>Raleigh, NC 27604</i> <i>(919) 790-0000</i> <i>Fax: (919) 790-0100</i>	<i>RevEng</i> is a family of intuitive, 3D digitising software for reverse engineering. RevEng dynamically links CMMs, 3D digitizers, & laser scanners to AutoCAD, CADKEY, & Pro-E to create 2D/3D CAD features & NURBS surfaces interactively from digitised points or random point cloud data.
	<i>RevSCAN</i> Surface Scanning
	<i>RevSURF</i> Surface Creation.
<i>Faro Technologies</i> <i>Lake Mary, FL</i> <i>407-333-9911</i>	<i>AnthroCam</i> CAD-based companion software for the FaroArm for Reverse Engineering and CAD-to-Part inspection
<i>Imageware</i>	<i>Surfacer</i> provides a complete software environment for working with 3D point data in conjunction with Computer Aided Geometric Design (CAGD) applications. It allows you to input geometry in the form of digitised point data, curves, or surfaces, to manipulate and analyse the geometry, and to output geometry to downstream analysis, design, visualisation, animation, or manufacturing processes. The Advanced Surfacing Module is available as an add-on to Surfacer v6.0. This new technology profoundly simplifies the creation of complex, multi-surface NURBS models from measurement data by drastically reducing the skill of the user as the constraint in the modelling process.



	<p><b>RPM Rapid Toolkit</b> was designed specifically with input from industry experts to meet the real-world needs of rapid prototyping users. RPM provides a wide range of tools, in an easy-to-use graphical environment, for producing and manipulating STL files.</p>
	<p><b>Verdict</b> is an innovative tool for 100% geometric verification of your part or tool. Verdict compares the actual part to the nominal design.</p>
<i>InnovMETRIC</i>	<p><b>POLYWORKS</b> is the first complete and fully integrated line of software tools for creating colour 3D models from 3D digitizer data organised as 3D meshes (such as produced by Cyberware or Moiré digitizers). With the highly automated POLYWORKS running on Silicon Graphics platforms, it takes less than a day to digitise and create an accurate polygonal model at several levels of resolution. Producing models directly in polygonal format (triangulation), POLYWORKS is good for real-time 3D graphics applications such as visualisation, visual simulation, creation of virtual environments, creation of virtual 3D collections, and multimedia.</p>
	<p><b>IMEdit/IMCompress</b> IMEdit is a semi-automated polygon editing toolbox which detects and repairs surface anomalies of 3D polygonal models. MCompress offers a surface-based algorithm to remove redundant information in large 3D polygonal models, for faster display rates. IMCompress is the only commercially available algorithm that guarantees true 3D tolerances between compressed and original models.</p>
<i>Laser Design</i>	<p><b>DATASCULPT 4.0</b> 3D Data Editing Software is a</p>

<i>Minneapolis, MN</i> <i>612-884-9648</i>	powerful point data editing software package which enables users to easily manipulate huge scan data files. It can manually or automatically process high- or low-density, 2D or 3D coordinate point data. "Point cloud" or more structured scan data are improved, reduced, merged, reorganised, interpolated, and edited using many CAD-like features.
<i>Raindrop Geomagic Inc.</i>	<i>Wrap</i> is a software product for automatic surface reconstruction and grid generation from point cloud data. The software can "wrap" a surface an arbitrary point cloud.

(Source: <http://www.geocities.com/CapeCanaveral/Lab/2549/reverse.htm#scan>)

**(b) Service Providers**

<i>Company</i>	<i>Description</i>
<i>3D Solution Lippstadt, Germany tel. + 44 (0) 2941 72440 fax + 44 (0) 2941 72441</i>	3D Solutions provides 3D Scanning, 3D Modelling, Animation/Rendering, Industrial Inspection, RE, Systems Sales, Systems Integration, Implementation & Training and Project Management.
<i>DesignCraft Solutions</i>	DesignCraft provides an array of rapid prototyping and modelling solutions and have a 3D Digitizer.
<i>GKS Inspection Services</i>	G.K.S. Inspection Services provide with information on products and services that will solve problems involving 3D inspection, digitising, RE, calibration, certification and all other facets of manufacturing that have to do with 3D metrology. Once aware of the technological capabilities that exist, G.K.S. assist in implementing these technologies in work environment via products and/or services.
<i>N Vision Inc.</i>	NVision offers the sale of the Digibot II as well as 3D modelling, RE and RP services.
<i>Scientific Measurement Systems, Inc.</i>	SMS X-ray tomography CT machines can scan metal, plastic, fiber or ceramic parts for use in 3D modelling, animation and engineering applications. SMS offers the sale of the SMS scanners as well as service for 3D modelling, RE and rapid prototyping services.
<i>Western Data Technologies, Inc.</i>	Western Data offers 3D laser scanning, 3D probe digitising, 3D CMM RE, 3D modelling services and the sale of both 3D digitising H/W and S/W.

(Source: <http://www.geocities.com/CapeCanaveral/Lab/2549/reverse.htm#scan>)



(c) Companies dealing with RE Technology

	Touch-Sensor Equipment Supplier		Optical Scanning Equipment Supplier		Touch-Sensor Equipment Supplier		Optical Scanning Equipment Supplier		RE S/W Manufacturer		RE S/W Distributor		Company Details	Product Description
<i>ADF Models-En Vormmakerij</i> <i>Wilrijk, Belgium</i> <i>tel. + 32 3 895 1921</i> <i>fax + 32 3 897 8380</i>				*										
<i>Arrk Europe (Deutschland) GmbH</i> <i>Deetzenbach-Steinberg,</i> <i>Germany</i> <i>tel. + 49 6074 37510</i> <i>fax + 49 6074 375123</i>				*		*								
<i>Arrk Europe France Sarl</i> <i>Colomiers, France</i> <i>tel. + 33 5 61 15 11 55</i> <i>fax + 33 5 61 15 14 81</i>				*		*								
<i>Arrk Europe France Sarl</i> <i>Orsay Cedex, France</i> <i>tel. + 33 1 60 19 33 72</i> <i>fax + 33 1 60 19 33 84</i>				*		*								
<i>Arrk Europe Ltd, London, UK</i> <i>tel. + 44 (0)181 961 6366</i> <i>fax + 44 (0)181 965 0201</i>				*		*								
<i>ARRK Europe Ltd, Milan, Italy</i> <i>tel. + 39 2 467 12253</i> <i>fax + 39 2 480 13233</i>				*		*								
<i>BCT GmbH, Dortmund, Germany</i> <i>tel. + 49 231 9750100</i>	*	*	*	*	*	*	*	*	*	*				<i>scancad:scan, laser,</i>

<i>fax + 49 231 97501099</i>							<i>range, mill, geo</i>
<i>CAD/CAM-concept GmbH Berlin, Germany tel. + 49 (0)30 63926315 fax + 49 (0)30 63926319</i>					*		<i>InterfaceTool, Corekt</i>
<i>Cadform Engineering GmbH Homburg/Ohm, Germany tel. +49 (0) 6633 977 0 fax +49 (0) 6633 977 610</i>			*	*			
<i>Cadtek Systems Ltd High Peak, UK tel. +44 (0)1663 741405 fax +44 (0)1663 741605</i>						*	<i>Matra Datavision, Strim</i>
<i>Cardiff Rapid Prototyping Centre Cardiff, UK tel. +44 (0)1222 874641 fax +44 (0)1222 874003</i>			*				
<i>CDI Technologies Ltd Leamington Spa, UK tel. +44 (0) 1926 431770 fax +44 (0) 1926 431880</i>	*	*					
<i>Cenit GmbH, Stuttgart, Germany tel. +49 711 780 73201 fax +49 711 780 73166</i>			*		*	*	<i>CAT-RPT</i>
<i>Centre for Rapid Product Development Ellison Place, UK tel. +44 (0)191 227 3644 fax +44 (0)191 227 3854</i>			*				
<i>CGI Inc., Minneapolis, MN, USA tel. +1 612 937 2005 fax +1 612 937 3018</i>			*				
<i>cirp GmbH, Stuttgart, Germany tel. +49 711 788960 fax +49 711 7889650</i>					*	*	<i>Pohos</i>
<i>CRIF, Seraing, Belgium tel. +32 4 361 8700 fax +32 4 361 8702</i>			*				
<i>Delcam International plc Birmingham, UK tel. +44 (0)121 766 5544 fax +44 (0)121 766 5511</i>			*		*		<i>Digicam, Copycad</i>
<i>DIGI, LAB srl</i>							

<i>Aprilia, Italy</i> <i>tel. +39 6 923387</i> <i>fax +39 6 9200906</i>		*		*		*	<i>Surfacer</i>
<i>Fraunhofer Institute (FhG-IPA)</i> <i>Stuttgart, Germany</i> <i>tel. +49 (0)711 970-1825</i> <i>fax +49 (0)711 970-1004</i>		*	*	*	*		<i>GUGS, CATIA- OPTI(optimise surface), Scan2CAD, CATIA-Point to Surface</i>
<i>GMS Ltd</i> <i>Hartford, Northwich, UK</i> <i>tel. +44 (0)1606 301133</i> <i>fax +44 (0)1606 301166</i>						*	<i>CALL</i>
<i>Helsinki University of Technology;</i> <i>Lahti Centre, Lahti, Finland</i> <i>tel. +358 3 5250 00</i> <i>fax +358 3 5250 204</i>			*				
<i>HK Technologies Ltd</i> <i>Rugby, UK</i> <i>tel. +44 (0) 1788 577288</i> <i>fax +44 (0) 1788 562808</i>	*	*					
<i>INTItec GmbH Software und</i> <i>Fertigungstechnologie,</i> <i>Aachen, Germany</i> <i>tel. +49 (0) 241 96098-80</i> <i>fax +49 (0) 241 96098-82</i>			*	*	*		<i>INTIsurf, INTIcheck, INTIcad</i>
<i>ISP Hurco</i> <i>High Wycombe, Bucks, UK</i> <i>tel. +44 (0) 1494 442222</i> <i>fax +44 (0) 1494 443350</i>						*	<i>Metris - Reverse Engineering</i>
<i>Laser Lines Ltd</i> <i>Banbury, Oxon, UK</i> <i>tel. +44 (0) 1295 267755</i> <i>fax +44 (0) 1295 269651</i>		*					
<i>Leyshon Miller Industries</i> <i>Cambridge, OH, USA</i> <i>tel. +1 614 432 2969</i> <i>fax +1 614 432 8865</i>			*				
<i>Materialise Software</i> <i>Leuven, Belgium</i> <i>tel. +32 16 298364</i> <i>fax +32 16 298319</i>					*		<i>CT Modeller</i>
<i>Matra Datavision</i> <i>Coventry, UK</i> <i>tel. +44 (0)1203 693939</i> <i>fax +44 (0)1203 693920</i>					*		<i>STRIM - Reverse Engineering</i>



<p><i>MSC Ltd, Andover, UK</i>  <i>tel. +44 (0)1264 350022</i>  <i>fax +44 (0)1264 350319</i></p>	*	*	*	*		*	
<p><i>Newport Ltd, Newbury, UK</i>  <i>tel. +44 (0)1635 521757</i>  <i>fax +44 (0)1635 521348</i></p>		*					
<p><i>Northern Technologies</i>  <i>Nelson, Lancs, England</i>  <i>tel. 01282 695186</i>  <i>fax 01282 698197</i></p>			*				
<p><i>NRU Prö zisionstechnologie</i>  <i>Neukirchen, Germany</i>  <i>tel. +49 371 28 1550</i>  <i>fax +49 371 28 155 20</i></p>		*		*			
<p><i>Oxon Prototypes Ltd</i>  <i>Chalgrove, Oxon, UK</i>  <i>tel. +44 (0)1865 891442</i>  <i>fax +44 (0)1865 891502</i></p>			*				
<p><i>PDC, Swansea, UK</i>  <i>tel. +44 (0)1792 481157</i>  <i>fax +44 (0)1792 481156</i></p>			*				
<p><i>Rapitypes Ltd, Leicester, UK</i>  <i>tel. +44 (0)116 253 6591</i>  <i>fax +44 (0)116 253 9827</i></p>			*				
<p><i>Renfrew Stylengineering</i>  <i>Leicester, UK</i>  <i>tel. +44 (0)116 253 1961</i>  <i>fax +44 (0)116 253 9827</i></p>			*				
<p><i>Renishaw plc</i>  <i>Wotton-under-Edge, Gloucs, UK</i>  <i>tel. +44 (0)1453 524524</i>  <i>fax +44 (0)1453 524101</i></p>	*	*			*	*	<i>Tracecut</i>
<p><i>Ricardo Consulting Engineers</i>  <i>Shoreham-by-Sea, UK</i>  <i>tel. +44 (0)1273 455611</i>  <i>fax +44 (0)1273 464124</i></p>			*				
<p><i>RP Solutions, Coventry, UK</i>  <i>tel. +44 (0)1203 632120</i>  <i>fax +44 (0)1203 632131</i></p>				*			
<p><i>Symore, Leuven,</i>  <i>Belgium</i>  <i>tel. +32 16 298 268</i>  <i>fax +32 16 298 319</i></p>						*	<i>CT-Modeller</i>

<i>Technimold srl</i> <i>Genova, Italy</i> <i>tel. ++33 10 650 3640</i> <i>fax ++33 10 650 8702</i>	*		*				
<i>The RP&amp;T Centre</i> <i>Coventry, UK</i> <i>tel. +44 (0)1203 524722</i> <i>fax +44 (0)1203 524878</i>			*	*			
<i>Thomas M Ltd</i> <i>Potters Bar, UK</i> <i>tel. +44 (0)1707 665044</i> <i>fax +44 (0)1707 665708</i>		*					
<i>Topologies Ltd, Daventry, UK</i> <i>tel. +44 (0)1327 310333</i> <i>fax +44 (0)1327 311325</i>	*			*		*	
<i>Unigraphics Solutions, an EDS</i> <i>Company</i> <i>Hook, Hamps, UK</i> <i>tel. +44 (0)1256 742000</i> <i>fax +44 (0)1256 742990</i>						*	<i>Unigrapics</i>
<i>ZAP - Ludwig RPO GmbH</i> <i>Gladback, Germany</i> <i>tel. +49 (0)2043 944 149</i> <i>fax +49 (0)2043 944 485</i>	*	*					

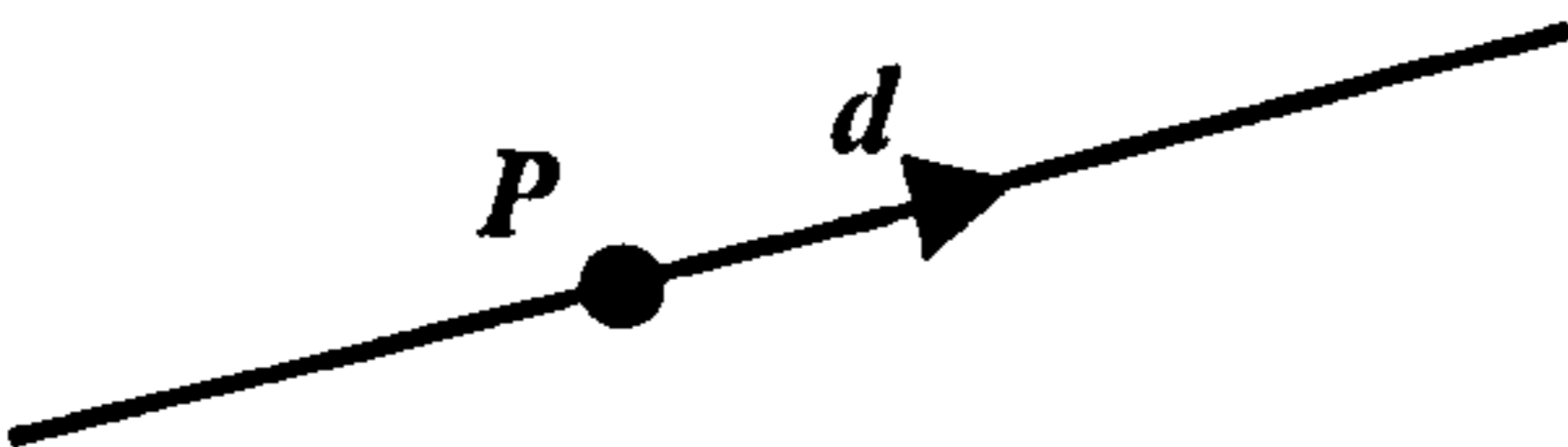
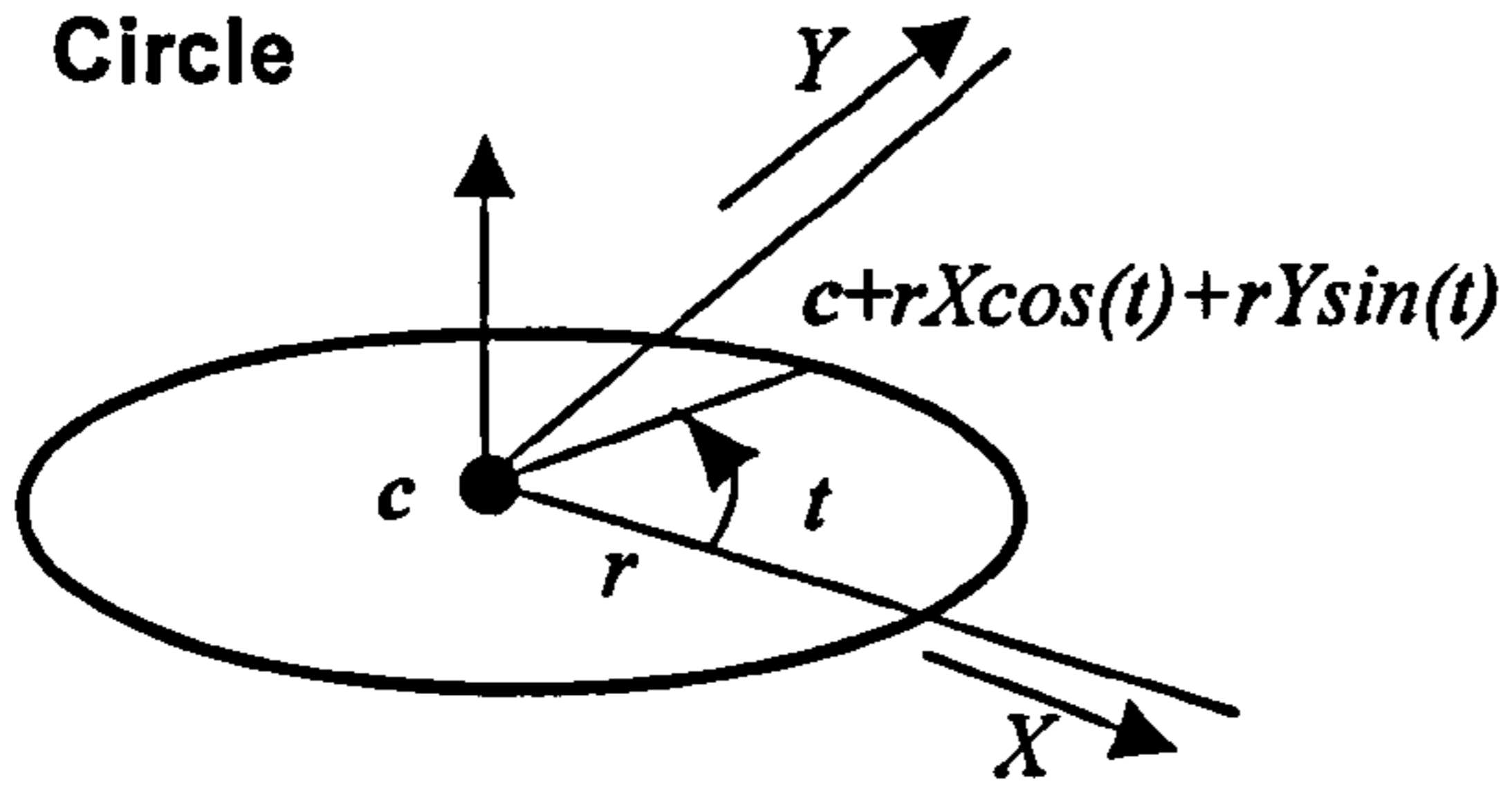
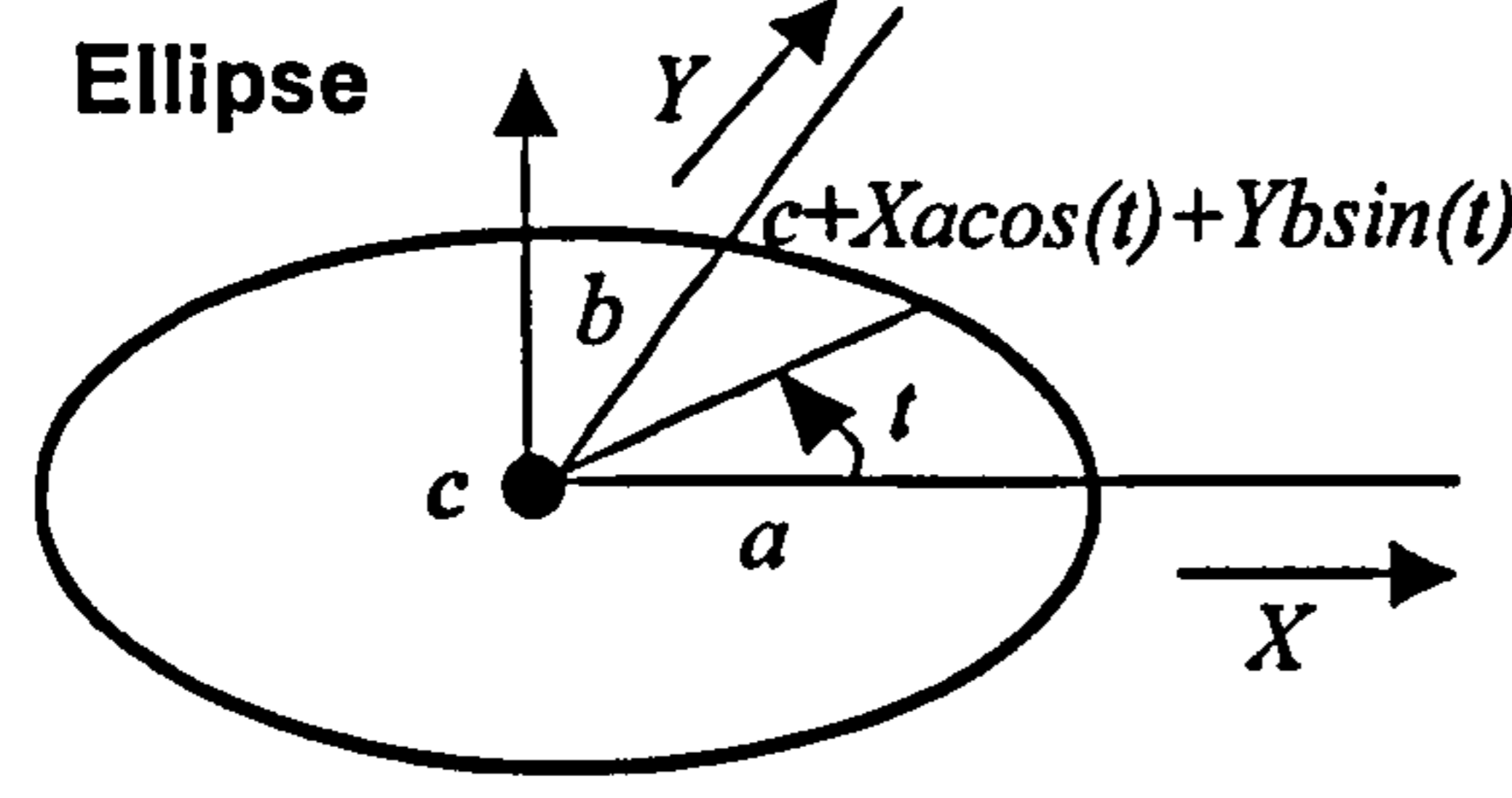
(Reproduced from *Time-Compression Technologies*, 6(1), January 1998, pp 68-70)

## Appendix 3

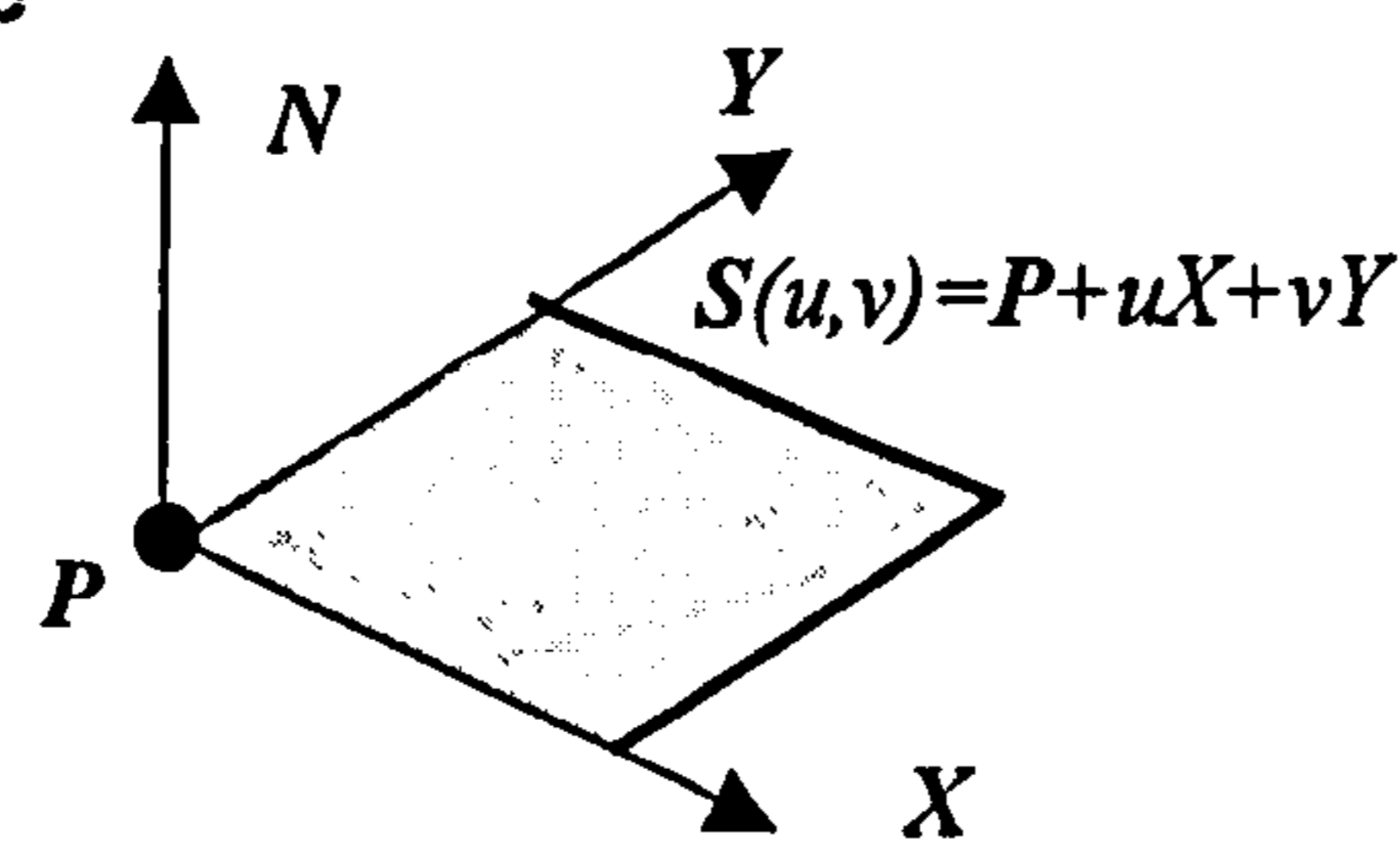
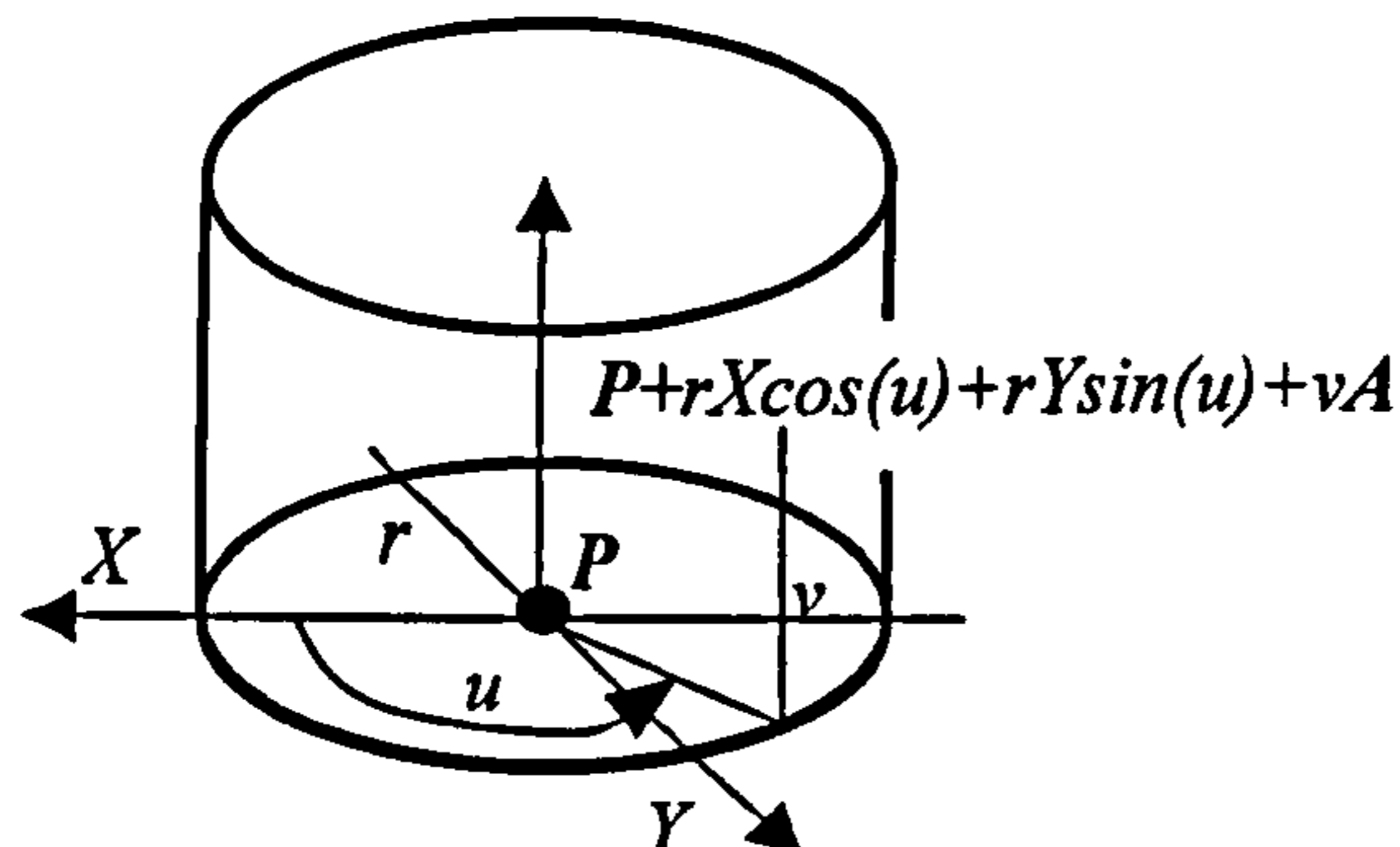
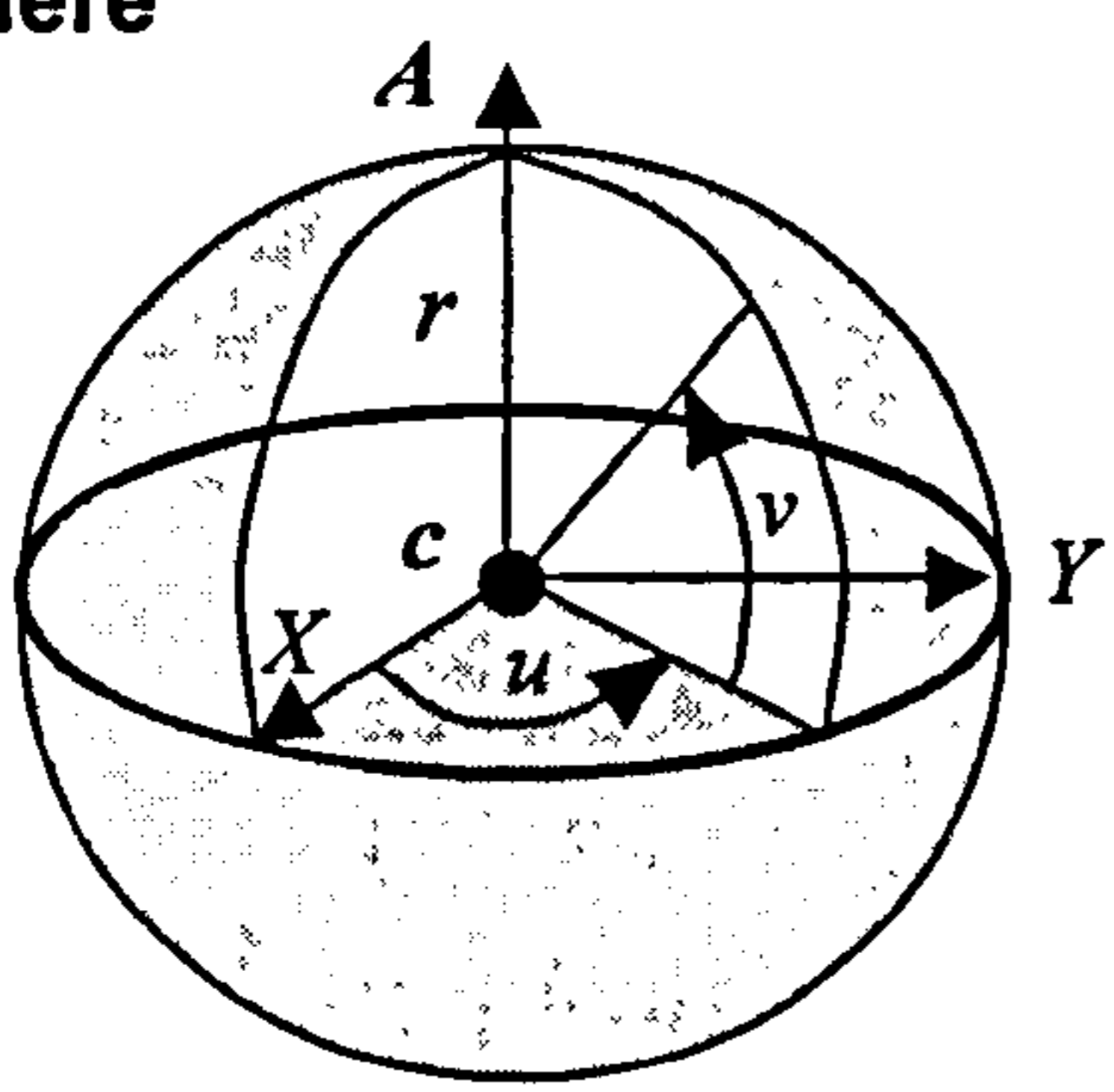
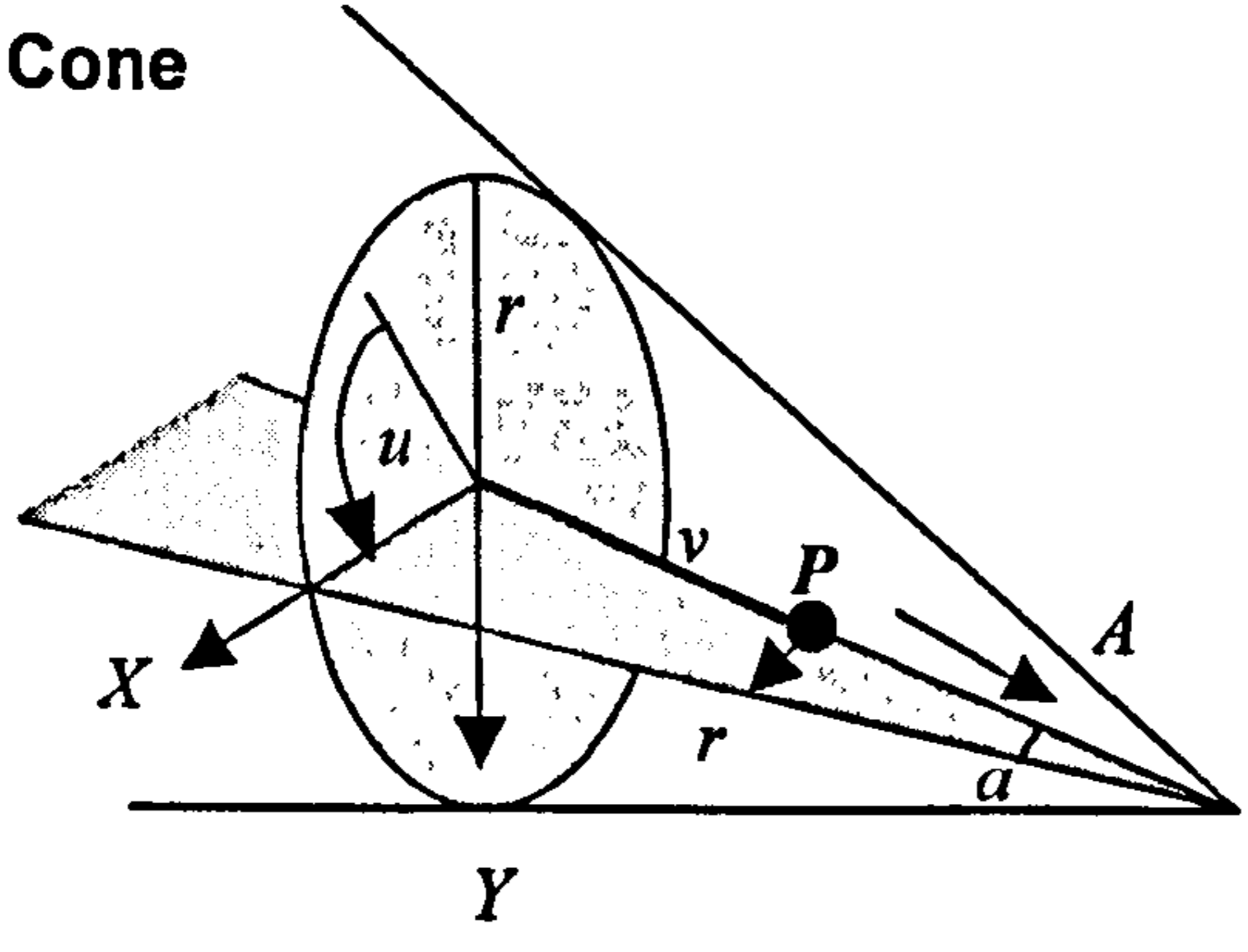
# Mathematical Representation for Analytic Entities

### (a) Parametric representation for analytic entities


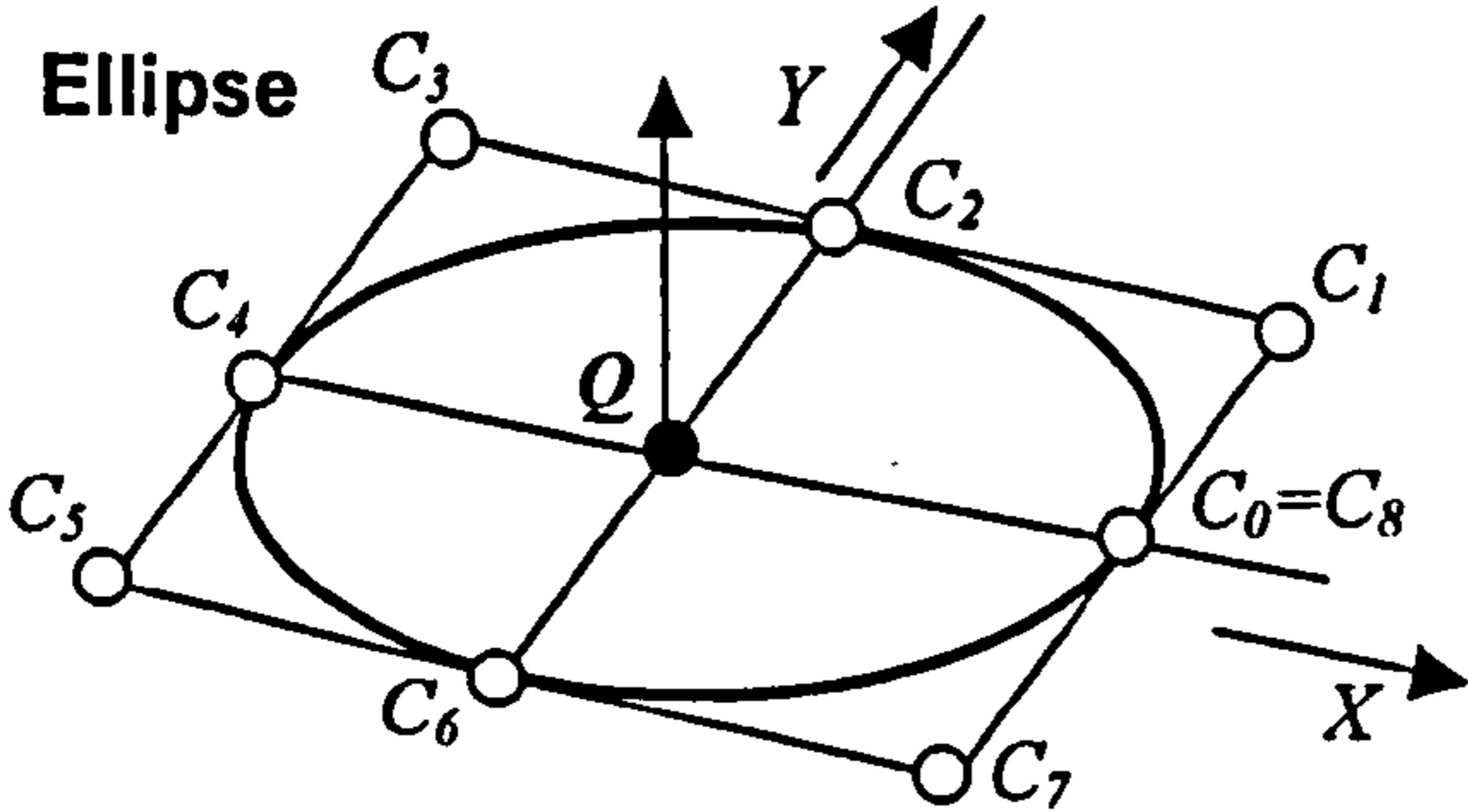
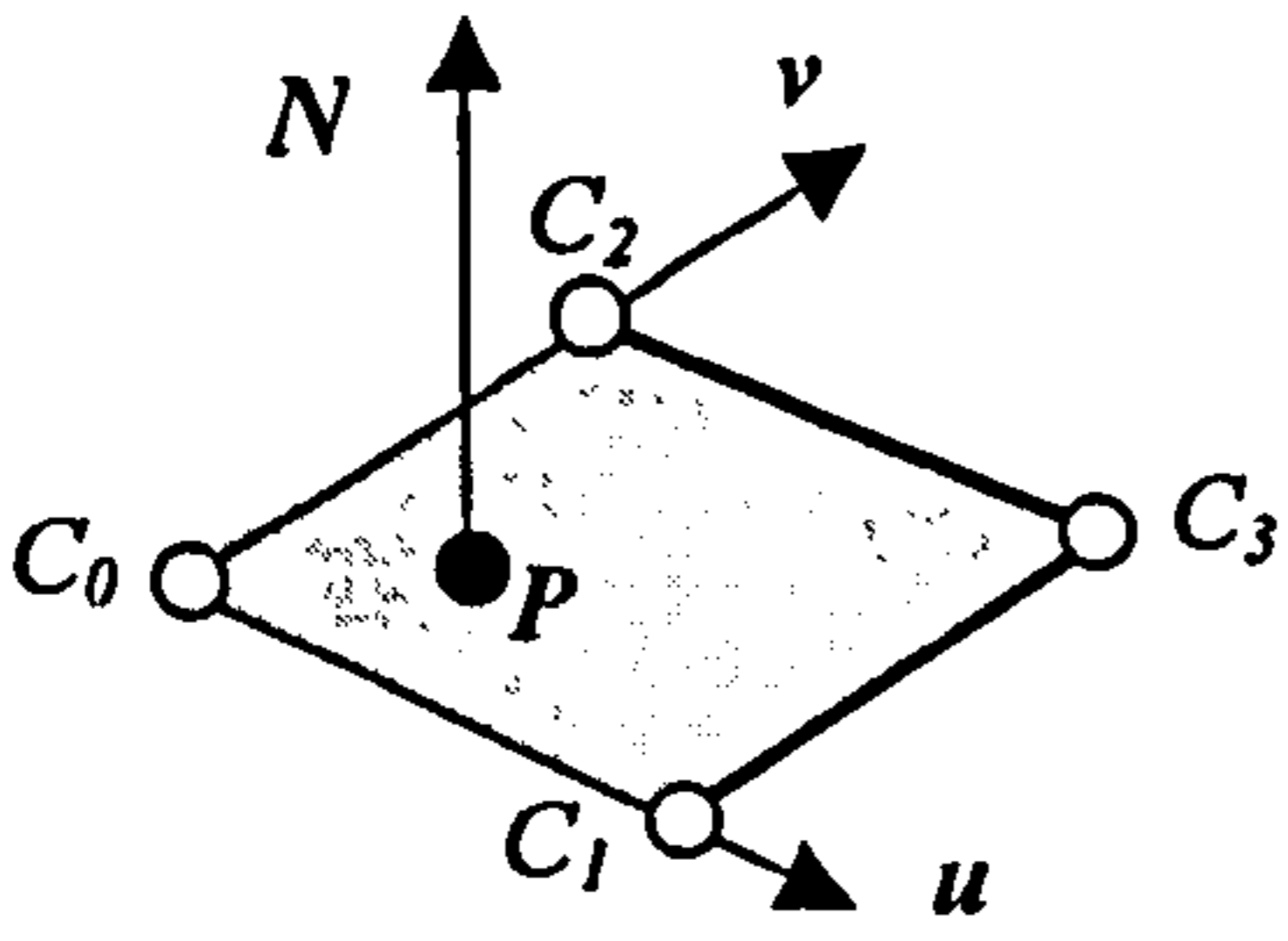
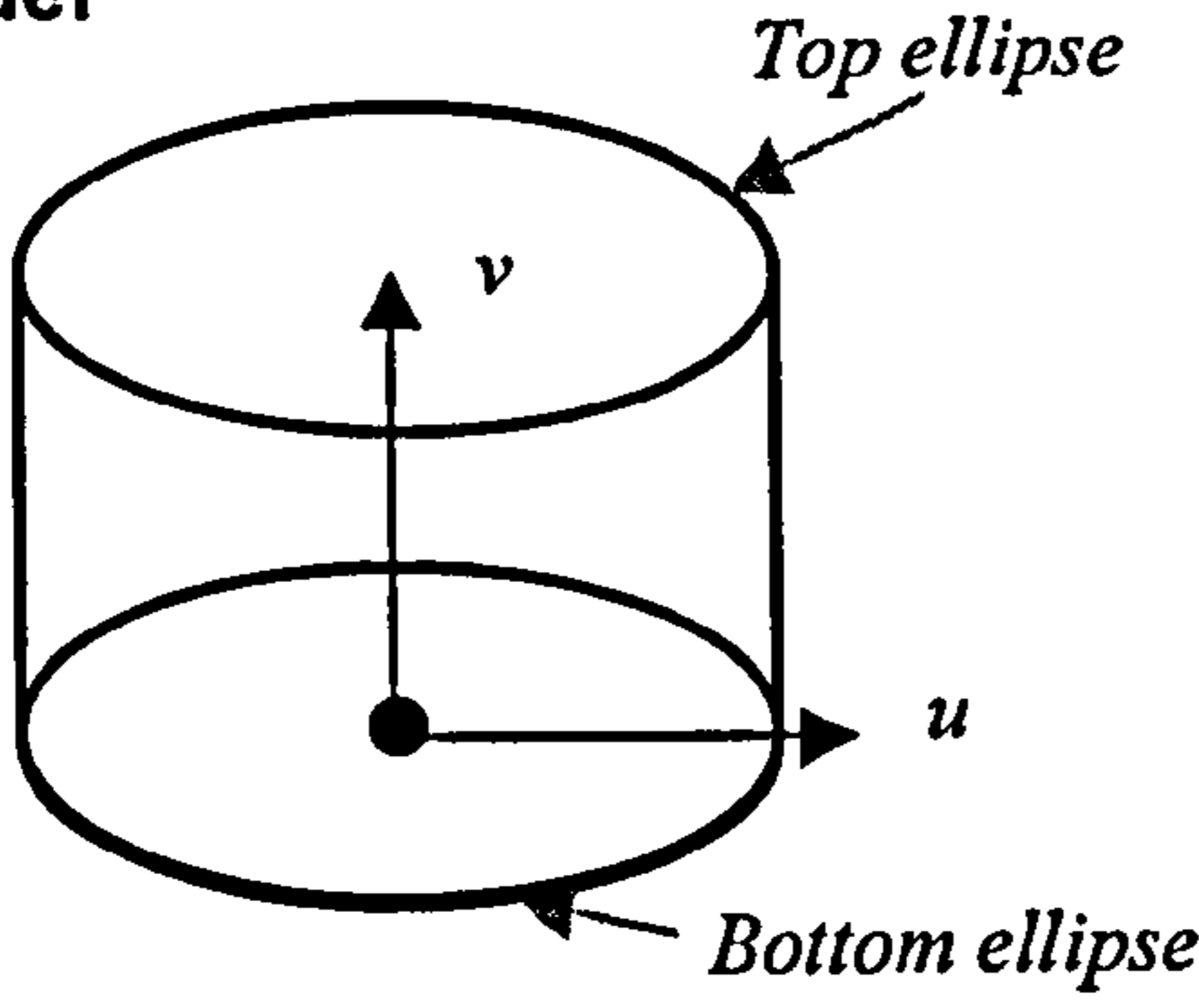
(Reproduced from Parasolid, 1998)

<p><b>Line</b></p>  <p>The diagram shows a straight line with a point labeled 'P' marked by a black dot. A vector labeled 'd' starts at 'P' and points along the line in the direction of the line's orientation.</p>	$R(t) = P + t d$ <p><math>P</math> is a point on the line.  <math>d</math> is the directional vector of line.          Parameter range: <math>0 \leq t \leq 1</math></p>
<p><b>Circle</b></p>  <p>The diagram shows an ellipse representing a circle. The center is labeled 'c'. Two axes, 'X' and 'Y', are shown as arrows originating from 'c'. A radius 'r' is shown as an arrow from 'c' to the ellipse. A point on the ellipse is labeled with the parametric equation <math>c + rX\cos(t) + rY\sin(t)</math>. An angle 't' is indicated between the radius and the X-axis.</p>	$R(t) = c + rX\cos(t) + rY\sin(t)$ <p><math>c</math> is the centre of the circle.  <math>r</math> is the circle radius.  <math>X</math> and <math>Y</math> are axes in the circle plane.          Parameter range: <math>0 \leq t \leq 2\pi</math></p>
<p><b>Ellipse</b></p>  <p>The diagram shows an ellipse. The center is labeled 'c'. Two axes, 'X' and 'Y', are shown as arrows originating from 'c'. The major axis is labeled 'a' and the minor axis is labeled 'b'. A point on the ellipse is labeled with the parametric equation <math>c + Xa\cos(t) + Ybsin(t)</math>. An angle 't' is indicated between the radius and the X-axis.</p>	$R(t) = c + aX\cos(t) + bY\sin(t)$ <p><math>c</math> is the centre of the ellipse.  <math>a</math> and <math>b</math> is the major and minor radius.  <math>X</math> and <math>Y</math> are major and minor axis in the ellipse plane.          Parameter range: <math>0 \leq t \leq 2\pi</math></p>



<p><b>Plane</b></p> 	$S(u,v) = P + uX + vY$ <p><math>P</math> is a point on the plane.  <math>X</math> and <math>Y</math> are axes in the plane.                      Parameter range: <math>-\infty \leq u, v \leq +\infty</math></p>
<p><b>Cylinder</b></p> 	$S(u,v) = P + rX\cos(u) + rY\sin(u) + vA$ <p><math>P</math> is the point on the cylinder axis.  <math>r</math> is the cylinder radius.  <math>A</math> is the cylinder axis.  <math>X</math> and <math>Y</math> are axes in the circle plane.                      Parameter range: <math>0 \leq u \leq 2\pi</math>,  <math>-\infty \leq v \leq +\infty</math></p>
<p><b>Sphere</b></p> 	$S(u,v) = c + (X\cos(u) + Y\sin(u))r\cos(v) + rA\sin(v)$ <p><math>c</math> is the centre of the sphere.  <math>r</math> is the sphere radius.  <math>A, X</math> and <math>Y</math> are ortho-normal axis.                      Parameter range: <math>0 \leq u \leq 2\pi</math>,  <math>-\frac{\pi}{2} \leq v \leq +\frac{\pi}{2}</math></p>
<p><b>Cone</b></p> 	$S(u,v) = P - vA + (X\cos(u) + Y\sin(u))(r + v \tan(a))$ <p><math>P</math> is the point on the cone axis.  <math>r</math> is the cone radius at the point <math>P</math>.  <math>a</math> is the cone half angle.  <math>A, X</math> and <math>Y</math> are ortho-normal axis.                      Parameter range: <math>0 \leq u \leq 2\pi</math>,  <math>-r \cos(a) \leq v \leq +\infty</math></p>

(b) B-spline representation for analytic entities (Park, 1997)

<p><b>Line</b></p> 	<p>Degree: 1                  Control points: <math>Sp</math> and <math>Ep</math>                  Knot vectors: 0, 0, 1, 1                  Weights: 1.0 at control points</p>
<p><b>Ellipse</b></p> 	<p>Degree: 2                  Control points: <math>C_0, C_1, \dots, C_8</math>                  Knot vectors: 0, 0, 0, 0.25, 0.25, 0.5, 0.5, 0.75, 0.75, 1, 1, 1                  Weights: 1.0, 0.7, 1.0, 0.7, 1.0, 0.7, 1.0, 0.7, 1.0</p>
<p><b>Plane</b></p> 	<p>U and V-degree: 1                  Control points: <math>C_0, C_1, C_2,</math> and <math>C_3</math>                  U-Knot vectors: 0, 0, 1, 1                  V-Knot vectors: 0, 0, 1, 1                  Weights: 1.0 at knot vectors</p>
<p><b>Cylinder</b></p> 	<p>U-degree: 2, V-degree: 1                  Control points: <math>C_0, C_1, \dots, C_8</math> on top ellipse and bottom ellipse                  Knot vectors: 0, 0, 0, 0.25, 0.25, 0.5, 0.5, 0.75, 0.75, 1, 1, 1                  Weights: 1.0, 0.7, 1.0, 0.7, 1.0, 0.7, 1.0, 0.7, 1.0, 1.0, 0.7, 1.0, 0.7, 1.0, 0.7, 1.0, 0.7, 1.0</p>

## Appendix 4

# Back-Propagation Algorithms

### (a) Training parameters

Before actual training starts, the following parameters must be provided into the ANN recogniser.

- Learning rate

This term is actually the rate of convergence between the current solution and the global minimum in the nature of the error space. Increasing learning rate as the network error decreases will often help to speed convergence by increasing the step size as the error reaches a minimum, but the network may bounce around too far from the global minimum value if learning rate gets too large (Freeman, 1992). The learning rate is used to determine the change of the current weights.

- Maximum root mean square error of output neurons

This term denotes the error criterion between the desired output and the actual output. If the difference between two output is less than this criterion, the training process is regarded as reaching the convergent state.



- Momentum rate

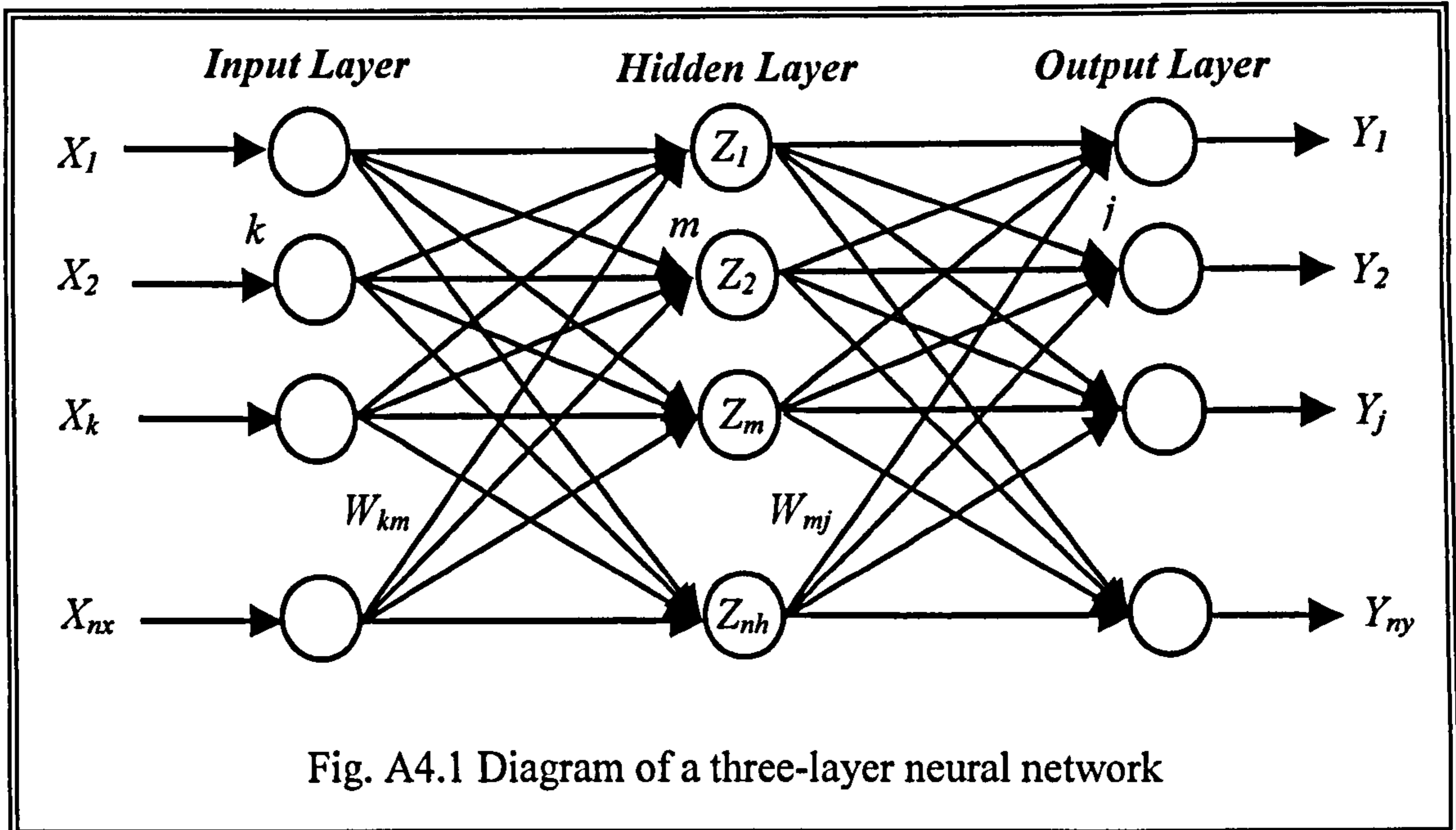
This term attempts to speed convergence at or near the global minimum by preventing the search from falling into local minima during the search process. If the total network error increases over a certain percentage from the previous iteration, the momentum rate is immediately set to zero until the error is again reduced. That is, this term tends to keep the weight changes going in the same direction. A momentum in the network is implemented by adding a fraction of the last weight change to the next one, that is  $\Delta W(t) + \mu\Delta W(t-1)$  which is used in the output and hidden layer, where  $t$  is a time step and  $\mu$  is the momentum. The momentum is a constant between 0.0 and 1.0. It multiplies the previous weight change to increase the likelihood of attaining the global optimum. In effect, the moment adjusts the weight changes so that the current solution bounces out of the local minimum and keeps moving in the direction of the global minimum.

- Bias term

The bias serves as a threshold, and its activation value is always 1.0.

**(b) BP training algorithms**

Figure A4.1 shows a three-layer network which was adopted in this application.



Each circle represents a neuron, and each arrow represents a connection and its associated weight. The input patterns are  $X_k$  ( $k = 1, 2, \dots, N_x$ ), the hidden layer neurons are  $Z_m$  ( $m = 1, 2, \dots, N_h$ ), the actual outputs are  $Y_j$  ( $j = 1, 2, \dots, N_y$ ), and the desired outputs are  $D_j$  ( $j = 1, 2, \dots, N_y$ ).

The BP training process is divided in two parts. In the first part, the weights are initialised to small random values with *zero mean*, and the network is then trained with training data corresponding to noise-free samples. The output neurons are monitored during training, so that, for any set of training patterns, the element of the output layer corresponding to that feature has to be high ( $\geq 0.95$ ) whilst the output of all other neurons has to be low ( $\leq 0.05$ ). For each kind of set of training data, the network needs to produce high activity in the

appropriate output neuron and low activity in the other neurons. In the second part of training is carried out with noisy training patterns, which nature of the noise is known. The training patterns of each feature are used in a straightforward manner to compute the weights corresponding to that feature. The training data pairs are created to cover the entire expected input space, and then, during the training process, they are randomly selected from the set. This is because the network will forget the initial training knowledge if the network is trained completely with a similar training pattern set, and then switch to different sets.

Due to this learning virtue, the BP neural network belongs to the category of supervised learning networks. On contrast, a neural network learning process is called unsupervised if its training set does not include desired outputs. This means that there is no guide used to check the error associated with actual outputs of the ANN.

The detailed BP training procedures are as following description:

- (STEP 1) Specify randomly the initial values of all weights to small real numbers. Input the value of the learning rate, the value of the momentum, and the value of the error tolerance.

An ANN is sensitive to the initial values of weights. Properly selected initial weights can shorten learning time and result in stable weights. Initial weights that are too small increase learning time, which may cause difficulties in converging to an optimal solution. On the contrary, if initial weights are too large, the weights may become unstable. In this research, the



initial weights were randomly selected between  $-0.25$  and  $0.25$ . The reason for this was that it simply works, and so far there is no systematic approach to determine the value of initial weights.

- (STEP 2) During feed-forward, each neuron ( $X_k$ ) in the input layer receives an input training pattern and computes its activation. Then it broadcasts the activation to the each of the hidden units  $Z_1, Z_2, \dots, Z_m$ .
- (STEP 3) Input layer communicates to hidden layer(s) where the actual processing is done. Each hidden neuron then computes its activation. The sums and activation values for each neuron in the hidden layer are propagated to the output layer to compute an actual output.
- (STEP 4) Each output neuron ( $Y_j$ ) computes its activation to form the current response of the network for the given input patterns. Actual outputs are generalised during the learning process, and desired outputs, which are regarded as a guide, are provided in the training set. Each output neuron compares its actual value  $Y_j$  with its desired value  $D_j$  to determine the associated error for that pattern with that neuron. The error  $E$  between desired and actual output is defined as  $E = \frac{1}{2} \sum_{j=1}^N (D_j - Y_j)^2$ , called *sum-squared error*. The equation is the object function where  $Y_j$  is a function of weights ( $W_{km}, W_{mj}$ ). By adjusting the weights in the neural network, the goal of the

learning process is to minimise this sum-squared error  $E$  over the entire training set and converge to the optimal solution.

- (STEP 5) The partial derivative of the sum-squared error of all neurons in the output layer is then propagated back to the hidden layer and the weights are adjusted. The factor  $\delta_j$  ( $j=1, 2, \dots, N_y$ ), called the error of neuron  $j$  in the output layer, is computed. It is used to update the weights between the output and the hidden layer. The value  $\delta_j$  has the following form:

$$\delta_j = -\frac{\partial E}{\partial i_m} = -\frac{\partial E}{\partial Y_j} \frac{\partial Y_j}{\partial i_m} = (D_j - Y_j) f'(i_m) = (D_j - Y_j) Y_j (1 - Y_j) \quad (\text{A4.1})$$

if  $\delta_j$  is within a predefined error tolerance, the training of the network is accomplished; otherwise the learning continues.  $\delta_j$  is used to distribute the error at output unit  $Y_j$  back to all units in the hidden layer. That is, the weights are modified by calculating and propagating the error of each neuron in the output layer backward through the entire network.

In a similar manner, it is also used to update the weights from the hidden layer to the input layer. The error factor  $\delta_m$  are computed for each hidden  $Z_m$  ( $m=1, 2, \dots, N_h$ ), that is calculate

$$\delta_m = Y_m (1 - Y_m) \sum_{j=1}^{N_y} W_{mj} \delta_j \quad (\text{A4.2})$$

- (STEP 6) The optimal solution is obtained by a partial derivative of the sum of the squared error with respect to the weights. The convergence toward

improved values of the weights and thresholds is accomplished by taking an incremental change  $\Delta W_{mj}$  proportional to  $\frac{-\partial E}{\partial W_{mj}}$ . Calculate the weight adjustments

$$\Delta W_{mj} = \eta \left( \frac{-\partial E}{\partial W_{mj}} \right) = \eta \left( \frac{-\partial E}{\partial i_m} \right) \frac{\partial i_m}{\partial W_{mj}} = \eta \delta_j \frac{\partial}{\partial W_{mj}} \left( \sum_{j=1}^{N_y} W_{mj} Y_j \right) = \eta \delta_j Y_j \quad (\text{A4.3})$$

where  $\eta$  is a constant called a *learning rate*, which is an adjustable step and is used to speed up convergence to a local minimum, and  $N_y$  denotes the number of neurons in the output layer. The final expression for computing the weight change and error in each neuron in the hidden layer are as follows:

$$\Delta W_{km} = \eta \delta_m Y_k \quad (\text{A4.4})$$

After all of the  $\delta$  factors have been determined, the weights for all layers are adjusted simultaneously

- (STEP 7) Repeat step 2 through 6 for next training pairs
- (STEP 8) Update new weights

$$W_{mj}(t+1) = W_{mj}(t) + \Delta W_{mj}(t) + \mu \Delta W_{mj}(t-1) \quad (\text{A4.5})$$

and

$$W_{km}(t+1) = W_{km}(t) + \Delta W_{km}(t) + \mu \Delta W_{km}(t-1) \quad (\text{A4.6})$$

where  $t$  is a time step



- (STEP 9) Repeat step 8 until the error for all vectors in the training set is reduced to an acceptable value

The training rule calculates small changes for a neuron's weights and biases in the direction that decreases the neuron's error. A trained ANN can be obtained only when the ANN algorithm yields a complete error-free iteration through all training patterns. Training patterns must be presented again to update the weights of connections until network learns all training sets. Once a neural network is trained, thereafter it may be used as an analytical tool to derive any meanings from complex or imperfect data in new and different circumstances. To do this, the user no longer specifies any training runs and instead allows the network to work in the forward propagation phase only. Even in the case that real input patterns have no relationship with the previous training patterns, it gives appropriate output with its taught knowledge.

The reasonable number of training pattern is proposed in reference (Freeman, 1992) as follows.

$$P = \frac{W}{e} \quad (\text{A4.7})$$

Where,  $P$ : number of training patterns  
 $W$ : number of weights to be trained  
 $e$ : the accuracy of classification expected.

## Appendix 5

# Mathematical Algorithms of MLE for Plane and Cylinder

### (a) Maximum-Likelihood Estimation (MLE) for a Plane

For a set of  $N$  scanned data points embedded in *3D Euclidean space* and assumed to be lying on a planar feature, the parameters of interest are  $\bar{p}$ , a point on the plane, and normal vector  $\bar{n}$ . The  $N$  scanned point  $Y = \{\bar{r}_i\}_{i=1}^N$  lying on this plane can be considered as random variables with the plane as the mean-value function and noise vectors perpendicular to the plane. Given that the noise vectors,  $\bar{n}'(\bar{r}_i - \bar{p})$ , are zero-mean *Gaussian* with variance  $\sigma^2$ , the likelihood of the scanned data is found from the product of the conditional likelihoods of each scanned point,

$$P(Y|\bar{n}, \bar{p}) = \prod_{i=1}^N \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2\sigma^2} \left(\bar{n}'(\bar{r}_i - \bar{p})\right)^2\right]$$

$$= (\sigma^2 2\pi)^{-\frac{N}{2}} \exp \left[ -\frac{1}{2\sigma^2} \bar{n}' \left\{ \sum_{i=1}^N (\bar{r}_i - \bar{p})(\bar{r}_i - \bar{p})' \bar{n} \right\} \right] \quad (\text{A5.1})$$

The MLE of  $\bar{p}$  and  $\bar{n}$  are found by minimising the negative logarithm of (A5.1).

The resulting equations are:

$$\frac{\partial}{\partial \bar{n}} \left\{ \sum_{i=1}^N \left[ \bar{n}' (\bar{r}_i - \bar{p}) \right]^2 + \lambda (\|\bar{n}\|^2 - 1) \right\} = 0 \quad (\text{A5.2})$$

$$\frac{\partial}{\partial \bar{p}} \left\{ \sum_{i=1}^N \left[ \bar{n}' (\bar{r}_i - \bar{p}) \right]^2 \right\} = 0 \quad (\text{A5.3})$$

The method of *Lagrange multipliers* has been used to include the unit vector constraint,

$\|\bar{n}\|^2 - 1 = 0$ . The MLE of  $\bar{p}$  from (A5.3) is

$$\bar{p} = \bar{r} = \frac{1}{M} \sum_{i=1}^N \bar{r}_i \quad (\text{A5.4})$$

This is clearly the mean vector  $\bar{r}$ . Equation (A5.2) is reduced to the expression for the MLE of the normal vector

$$\left[ \sum_{i=1}^N (\bar{r}_i - \bar{p})(\bar{r}_i - \bar{p})' \right] \bar{n} + \lambda \bar{n} = 0 \quad (\text{A5.5})$$



The normal vector  $\underline{n}$  is an *eigenvector* of the covariance matrix

$$Cov = \left[ \sum_{i=1}^N (\underline{r}_i - \bar{p})(\underline{r}_i - \bar{p})^T \right] \quad (A5.6)$$

To compute the parameter  $\bar{p}$  and  $\bar{n}$  of the plane, the mean vector  $\bar{r}$  and covariance matrix,  $Cov$ , of the data set must be calculated. The mean vector is calculated from (A5.4) and *Jacobi method* for finding the eigenvalues and eigenvectors of a real symmetric matrix is used to determine the eigenvalues  $\lambda_1 \geq \lambda_2 \geq \lambda_3$  and the eigenvectors  $\underline{v}_1$ ,  $\underline{v}_2$  and  $\underline{v}_3$  of the covariance matrix (A5.6). The eigen vectors  $\underline{v}_1$  and  $\underline{v}_2$  define a plane parallel to and passing through the data centroid whereas  $\underline{v}_3 = (v_{31}, v_{32}, v_{33})$  is the vector normal to that plane. Given the equation of a plane

$$ax + by + cz + d = 0 \quad (A5.7)$$

where the vector  $(a, b, c)$  is the normal to the plane, the following equations can be made

$$a = v_{31} \quad b = v_{32} \quad c = v_{33} \quad d = a\underline{x} + b\underline{y} + c\underline{z} \quad (A5.8)$$

**(b) Maximum-Likelihood Estimation for a Cylinder**

For a cylinder, with  $\bar{n}$  the direction of the axis,  $\bar{p}$  a point on the axis, and  $Q$  the radius, the likelihood of a set of  $N$  scanned points  $Y = \{\bar{r}_i\}_{i=1}^N$  can be formulated. Employing the same noise model, the distance from a range point to the true cylinder axis has to be determined.

This distance is expressed as:

$$\sqrt{(\bar{r}_i - \bar{p})' [I - \bar{n}\bar{n}'] (\bar{r}_i - \bar{p})} = \|\bar{r}_i - \bar{p} - (\bar{r}_i - \bar{p})' \bar{n}\bar{n}'\| \quad (\text{A5.9})$$

Now the  $(\|\bar{r}_i - \bar{p} - (\bar{r}_i - \bar{p})' \bar{n}\bar{n}'\| - Q)$  are assumed to be zero-mean *Gaussian* and independent, yielding the likelihood of the data

$$\begin{aligned} P(Y|\bar{n}, \bar{p}, Q) &= \prod_{i=1}^N \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2\sigma^2} (\|\bar{r}_i - \bar{p} - (\bar{r}_i - \bar{p})' \bar{n}\bar{n}'\| - Q)^2\right] \\ &= (\sigma^2 2\pi)^{-\frac{N}{2}} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^N (\|\bar{r}_i - \bar{p} - (\bar{r}_i - \bar{p})' \bar{n}\bar{n}'\| - Q)^2\right] \\ &= (\sigma^2 2\pi)^{-\frac{N}{2}} \exp\left[-\frac{1}{2\sigma^2} \left\{ \sum_{i=1}^N \left( \sqrt{(\bar{r}_i - \bar{p})' [I - \bar{n}\bar{n}'] (\bar{r}_i - \bar{p})} - Q \right)^2 \right\} \right] \end{aligned} \quad (\text{A5.10})$$

Again the *MLE*'s  $\hat{n}$ ,  $\hat{p}$ , and  $\hat{Q}$  of the cylinder parameters may be determined by minimising

$$\sum_{i=1}^N (\|\bar{r}_i - \bar{p} - (\bar{r}_i - \bar{p})' \bar{n}\bar{n}'\| - Q)^2 \quad (\text{A5.11})$$

with respect to  $\bar{n}$ ,  $\bar{p}$ , and  $Q$ .

## Appendix 6

# IGES File Overview

IGES focuses on basic geometric model information and, despite its limitations, remains the dominant standard for geometry data exchange.

An IGES file is a sequential file consisting of a sequence of records. The file formats treat the product definition to be exchanged as a file of entities, each entity being represented in a standard format, to and from which the internal representation of a specific CAD/CAM system can be mapped. In the ASCII format, the entire file is of 80-character records. The file is divided into sections. Within each section, the records are labeled and numbered. IGES data is written in columns 1 through 72 inclusive each record. Column 73 stores the section identification character. Columns 74 through 80 are specified for the section sequence number of each record.

An IGES file consists of six sections which must appear in the following order: *Start section, Global section, Directory Entry section, Parameter Data section, and Terminate section.* Figure A10.1 shows an IGES ASCII file created by the proposed



application. It shows a various sections of a typical IGES file and details of each section.

A	ReTool IGES file								S	1
B	1H,,1H;,4Hface,55HC:\PrograFiles\DevStudio\MyProjects\Replica\a_GOD.igs,G								G	1
	34HClone by Warwick Manufacturing Grp,7Hunnamed,32,38,7,								G	2
	38,15,4Hface,1.,1,4HINCH,32768,0.5,13H990028.135632,0.000561065,								G	3
	5.61087,3Hjun,27HWarwick Manufacturing Group,6,0;								G	4
C	128	1	1	1	0	0	0	001010000D	1	
	128	0	0	32	0	0	0	MIPTR 1D	2	
	128	32	1	1	0	0	0	001010000D	3	
	128	0	0	28	0	0	0	MIPTR 2D	4	
	128	60	1	1	0	0	0	001010000D	5	
	128	0	0	27	0	0	0	MIPTR 3D	6	
	128	87	1	1	0	0	0	001010000D	7	
	128	0	0	23	0	0	0	MIPTR 4D	8	
	128	110	1	1	0	0	0	001010000D	9	
	128	0	0	23	0	0	0	MIPTR 5D	10	
.										
.										
.										
.										
128	987	1	1	0	0	0	0	001010000D	63	
128	0	0	34	0	0	0	0	MIPTR 32D	64	
128	1021	1	1	0	0	0	0	001010000D	65	
128	0	0	58	0	0	0	0	MIPTR 33D	66	
D	128,5,5,3,3,0,0,1,0,0,0.0D0,0.0D0,0.0D0,0.0D0,8.0D0,1.6D1,								1P	1
	2.4D1,2.4D1,2.4D1,2.4D1,0.0D0,0.0D0,0.0D0,0.0D0,								1P	2
	3.33333358764648D1,6.6666717529297D1,1.0D2,1.0D2,1.0D2,								1P	3
	1.0D2,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,								1P	4
	1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,								1P	5
	1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,								1P	6
	1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D2,5.0D1,0.0D0,								1P	7
	1.0D2,5.0D1,0.0D0,1.0D2,5.0D1,-9.59999942779541D0,								1P	8
.										
.										
.										
.										
.-1.31609182357788D1,7.5D1,1.8D1,-1.38199996948242D1,7.5D1,								65P	1072	
1.8D1,-1.38199996948242D1,7.5D1,2.19828128814697D1,								65P	1073	
-1.38200006484985D1,7.50000076293945D1,2.50000038146973D1,								65P	1074	
-1.38200016021729D1,7.5D1,2.80171890258789D1,								65P	1075	
-1.38199996948242D1,7.5D1,3.2D1,-1.38199996948242D1,7.5D1,								65P	1076	
3.2D1,-1.38199996948242D1,0.0D0,1.40445022583008D1,0.0D0,								65P	1077	
5.02137603759766D1;								65P	1078	
E	S	1G	4D	66P	1078				T	1

Fig. A6.1 A sample ASCII IGES file. (A) Start section, (B) Global section, (C) Directory Entry section, (D) Parameter Data section, (E) Terminate section

The *Start section* is a human-readable prologue to the file such as the names of the source and target CAD/CAM systems, and a brief description of the product being converted. There must be at least one start record. All records in the section shall have the letter S in column 73 and a sequence number in column 74 through 80.

The *Global section* includes information describing the pre-processor and information needed by the postprocessor to interpret the file. All records in the global section shall contain the letter G in column 73 and a sequence number. The parameters for the global section are input in free format.

The contents defined in this section are:

- parameter delimiter (,)
- record delimiter (;)
- product name of sending or originating CAD system
- IGES file name
- system IS
- reference of IGES processor
- Number of bit for integer representation
- maximum power of ten representable in a single precision floating number on the sending system
- Number of digits
- power of ten for double
- Digit for double
- product ID for receiving system
- model space scale
- unit flag
- unit
- maximum number of line weight gradation

- size of maximum line width
- date and time of generation
- minimum granularity expressed in unit
- maximum coordinate value in units
- name of author
- organisation
- IGES version
- drafting standard

The *DE section* for each entity is fixed in size and contains twenty fields of eight characters each spread across two consecutive eighty character lines. The purposes of the directory section are to provide an index for the file and to contain attribute information for each entity. Some of the entity attributes defined in this section are line font, layer number, transformation matrix, line weight, and colour. A listing of the fields making up the directory entry for each entity is shown in Table A6.1.

Table A6.1. Format of the Directory Entry (DE) section in the *ASCII* IGES form.

1	8	9	16	17	24	25	32	33	40	41	48	49	56	57	64	65	72	73	80
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	Entity Type Number	Parameter Data	Structure	Line Font Pattern	Level	View	Transfor- mation Matrix	Label Display Assoc.	Status Number	Sequence Number
(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)	Entity Type Number	Line Weight Number	Colour Number	Parameter Line Count	Form Number	Reserved	Reserved	Entity Label	Entity Subscript Number	Sequence Number



The *PD section* contains the actual data defining each entity listed in the DE section. For instance, a line entity is specified by the six coordinates of its two end points. Whilst each entity has always two records in the DE section, the number of records needed for each in the PD section varies from one entity to another and depends on the amount of data. For instance, the PD section of rational B-spline surface entity (type 128) is as follows.

$$Q \approx S(u, v) = \frac{\sum_{i=0}^{K1} \sum_{j=0}^{K2} W_{i,j} V_{i,j} N_{i,M1}(u) N_{j,M2}(v)}{\sum_{i=0}^{K1} \sum_{j=0}^{K2} W_{i,j} N_{i,M1}(u) N_{j,M2}(v)}$$

where  $S(u, v)$  is the position vector along the surface as a function of the parameters  $u$  and  $v$ ;  $V_{ij}$  are the  $(K1+1)(K2+1)$  control point net for the patch;  $N_{i,M1}(u)$  and  $N_{j,M2}(v)$  are the B-splines basis functions of degree  $M1$  and  $M2$  in the  $u$  and  $v$  directions defined over suitably chosen knot vectors. The  $W_{ij}$  are the weights.

$$\text{Here, } N1 = K1 - M1 + 1$$

$$N2 = K2 - M2 + 1$$

$$A = N1 + 2M1$$

$$B = N2 + 2M2$$

$$C = (K1+1)*(K2+1)$$

Detailed parameter data of B-spline surface is depicted in Table A6.2.

Table A6.2. The PD section of B-spline surfaces.

INDEX	NAME	TYPE	DESCRIPTION
1	<i>K1</i>	<i>int</i>	<i>Upper index of first sum</i>
2	<i>K2</i>	<i>int</i>	<i>Upper index of second sum</i>
3	<i>M1</i>	<i>int</i>	<i>Degree of first set</i>
4	<i>M2</i>	<i>int</i>	<i>Degree of second set</i>
5	<i>Prop1</i>	<i>int</i>	<i>1 = Closed in u-direction, 0 = Not closed</i>
6	<i>Prop2</i>	<i>int</i>	<i>1 = Closed in v-direction, 0 = Not closed</i>
7	<i>Prop3</i>	<i>int</i>	<i>1 = Rational, 0 = Polynomial</i>
8	<i>Prop4</i>	<i>int</i>	<i>1 = Non-periodic in u-dir, 0 = Periodic</i>
9	<i>Prop5</i>	<i>int</i>	<i>1 = Non-periodic in v-dir, 0 = Periodic</i>
<i>10 ~ 10+A</i>	<i>Uknot</i>	<i>real</i>	<i>first knot sequence</i>
<i>11+A ~ 11+A+B</i>	<i>Vknot</i>	<i>real</i>	<i>second knot sequence</i>
<i>12+A+B ~ 11+A+B+C</i>	<i>Weights</i>	<i>real</i>	<i>weights</i>
<i>12+A+B+C ~ 11+A+B+4*C</i>	<i>Ctrlpts</i>	<i>real</i>	<i>control points</i>
<i>12+A+B+4*C</i>	<i>U0</i>	<i>real</i>	<i>Starting value for u-direction</i>
<i>13+A+B+4*C</i>	<i>U1</i>	<i>real</i>	<i>Ending value</i>
<i>14+A+B+4*C</i>	<i>V0</i>	<i>real</i>	<i>Starting value for v-direction</i>
<i>15+A+B+4*C</i>	<i>V1</i>	<i>real</i>	<i>Ending value</i>

The *Terminate* section contains a single record which specifies the number of records in each of the four preceding sections for checking purpose.