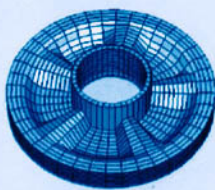
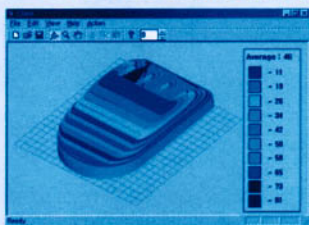
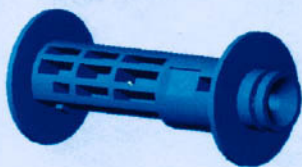


# Software Solutions for Rapid Prototyping



**Edited by  
Ian Gibson**



**Professional  
Engineering  
Publishing**

# Software Solutions for Rapid Prototyping

*This page intentionally left blank*

# Software Solutions for Rapid Prototyping

Edited by I Gibson



Professional Engineering Publishing Limited  
London and Bury St Edmunds, UK



First published in 2002 by Professional Engineering Publishing Limited, UK.

This publication is copyright under the Berne Convention and the International Copyright Convention. All rights reserved. Apart from any fair dealing for the purpose of private study, research, criticism or review, as permitted under the Copyright, Designs and Patents Act, 1988, no part may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, electrical, chemical, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owners. Unlicensed multiple copying of the contents of this publication is illegal. Inquiries should be addressed to: The Publishing Editor, Professional Engineering Publishing Limited, Northgate Avenue, Bury St Edmunds, Suffolk, IP32 6BW, UK.

© 2002 I Gibson and contributing authors

ISBN 1 86058 360 1

A CIP catalogue record for this book is available from the British Library.

Printed and bound in Great Britain  
by The Cromwell Press Limited, Wiltshire, UK.

The Publishers are not responsible for any statement made in this publication. Data, discussion, and conclusions developed by the Author are for information only and are not intended for use without independent substantiating investigation on the part of potential users. Opinions expressed are those of the Author and are not necessarily those of the Institution of Mechanical Engineers or its Publishers.

# Related Titles

<i>Advances in Manufacturing Technology XV</i>	D T Pham, S S Dimov, and V O'Hagan	1 86058 325 3
<i>Computer-aided Production Engineering CAPE 2001</i>	H Bin	1 86058 367 9
<i>Developments in Rapid Prototyping and Tooling</i>	G Bennett	1 86058 048 3
<i>3rd National Conference on Rapid Prototyping, Tooling, and Manufacturing</i>	A E W Rennie, D M Jacobson, and C E Bocking	1 86058 374 1
<i>Rapid Prototyping Casebook</i>	J A McDonald, C J Ryall, and D I Wimpenny	1 86058 076 9
<i>Quality, Reliability, and Maintenance QRM 2002</i>	G J McNulty	1 86058 369 5

For the full range of titles published by Professional Engineering Publishing contact:

Marketing Department  
Professional Engineering Publishing Limited  
Northgate Avenue, Bury St Edmunds  
Suffolk, IP32 6BW  
UK  
Tel: +44(0)1284 724384 Fax: +44(0)1284 718692  
email: [orders@pepublishing.com](mailto:orders@pepublishing.com)  
[www.pepublishing.com](http://www.pepublishing.com)

*This page intentionally left blank*

# Contents

	<b>Introduction</b>	<b>1</b>
<b>Chapter 1</b>	<b>Advanced pre-processor functionality for layered manufacturing</b>	<b>7</b>
1.1	Introduction	8
1.2	Functionality of commercial LM software	9
1.2.1	Viewing/editing	9
1.2.2	Repair/optimization	9
1.2.3	Layout/conversion	9
1.2.4	Example: three-dimensional-systems stereolithography software	11
1.3	Advanced functionality of LM pre-processor	15
1.3.1	Intelligent pre-processor based on design visualization	15
1.3.2	Distributed pre-processor based on network	18
1.4	Distributed intelligent pre-processor	21
1.4.1	System environment	21
1.4.2	System implementation	22
1.5	Conclusions	23
<b>Chapter 2</b>	<b>A web-based database system for RP machines, processes, and materials selection</b>	<b>27</b>
2.1	Introduction	27
2.1.1	RP website	28
2.1.2	Objectives	30
2.1.3	Development backgrounds	31
2.2	Methodology	31
2.2.1	RP website design	32
2.2.2	RP data collections	41
2.2.3	Copyright issues	42
2.2.4	Selection of web-based design tools	42
2.2.5	Choosing a database system	43

2.3	System implementation	43
2.3.1	RP decision support system main page	43
2.3.2	RP product information main page	44
2.3.3	Browsing of RP machine data	45
2.3.4	Register as a new RP vendor	47
2.3.5	Logging in as RP vendors	48
2.3.6	Add new RP machine information	48
2.3.7	Update RP machine information	50
2.3.8	Update RP company information	50
2.4	An extended database system for RP decision support	50
2.4.1	The methodology of DSS	53
2.4.2	Use of the database	54
2.4.3	Scope of future work	54
<b>Chapter 3</b>	<b>Rapid prototyping and manufacturing (RP&amp;M) benchmarking</b>	<b>57</b>
3.1	Introduction	58
3.2	Review of RP&M benchmark parts	59
3.2.1	Kruth 1991	59
3.2.2	Gargiulo – 3D Systems 1992	60
3.2.3	Lart 1992	60
3.2.4	Juster and Childs (1994)	60
3.2.5	Ippolito, Iuliano, and Filippi 1994	61
3.2.6	Ippolito, Iuliano, and Gatto 1995	62
3.2.7	Shellabear – EOS Gmbh 1998, and Reeves and Cobb	62
3.2.8	Fen and Dongping 1999	64
3.3	Types of RP&M benchmarks	64
3.3.1	Geometric benchmark	65
3.3.2	Mechanical benchmark part	66
3.3.3	Key attributes for benchmark parts	68
3.4	Towards generalized benchmark parts	68
3.4.1	Proposed geometric benchmark	69
3.5	Process benchmarks	74
3.5.1	RP&M process chains	75
3.5.2	Measurement of RP&M parts	76
3.5.3	Considerations for process benchmarking	76
3.6	Framework for process benchmarking	79
3.7	RP&M material selection	80

3.8	Integration with database for decision support	80
3.9	Conclusion	82
	Appendix A3	85
A3.1	Key considerations in process chains	85
	A3.1.1 Orientation	85
	A3.1.2 Slicing	85
	A3.1.3 Supported/supportless building	86
A3.1.4	Support structures	87
	A3.1.5 Path planning	87
	A3.1.6 Pre-processing	88
	A3.1.7 Post-processing	88
	A3.1.8 Process parameters	89
A3.2	Weakness/difficulties	90
	A3.2.1 Surface roughness	90
	A3.2.2 Material shrinkage or swelling	90
	A3.2.3 Staircase effect	90
	A3.2.4 Warpage	91
	Appendix B3	92
<b>Chapter 4</b>	<b>Decision support and system selection for RP</b>	<b>95</b>
4.1	Introduction	95
4.2	Selection methods for a part	97
	4.2.1 Approaches to determining feasibility	98
	4.2.2 Approaches to selection	99
4.3	A rapid prototyping system selection guide	111
	4.3.1 Expert system selection	113
	4.3.2 Rules and weighting system	114
	4.3.3 Example web pages and search process	116
	4.3.4 Connection to database	120
	4.3.5 Further modification of the system	120
4.4	Production planning and control	121
	4.4.1 Pre-processing	122
	4.4.2 Part build	123
	4.4.3 Post-processing	123
	4.4.4 Summary	124
4.5	Open problems	124
<b>Chapter 5</b>	<b>Adaptive slicing for rapid prototyping</b>	<b>129</b>
5.1	Introduction	130
5.2	CAD/CAM/RP data processing	131

	5.2.1	Calculating the intersection points	132
	5.2.2	Sorting the intersection points	133
	5.2.3	Constructing a two-dimensional contour	133
	5.2.4	Generating support structure	134
5.3		Fundamentals of adaptive slicing	136
	5.3.1	Basic adaptive slicing	137
	5.3.2	Analyses for basic adaptive slicing	139
	5.3.3	Problems for current adaptive slicing	142
5.4		Advanced adaptive slicing	143
	5.4.1	Procedures for advanced adaptive slicing	144
	5.4.2	Analysis	148
5.5		Conclusion	153
<b>Chapter 6</b>		<b>Multi-material representation and design issues</b>	<b>155</b>
6.1		Introduction	155
6.2		Literature review	157
	6.2.1	Colour representation and colour schemes	158
	6.2.2	Geosciences modelling	161
	6.2.3	Volumetric representations	162
	6.2.4	Interpolation and implicit methods	168
	6.2.5	Previous work in multi-material/ heterogeneous solid representation	178
6.3		MMA-rep, a V-representation for multi- material objects	188
	6.3.1	Design philosophy	188
	6.3.2	Designer's knowledge of the problem	189
	6.3.3	Material distribution data	190
	6.3.4	Interpolation space and interpolation kernel	195
	6.3.5	Representation summary	198
6.4		Applications	199
	6.4.1	One-dimensional solids	199
	6.4.2	Two-dimensional solids	209
	6.4.3	Three-dimensional solids	216
	6.4.4	Summary of MMA-rep features	218
6.5		Conclusion	219

<b>Chapter 7</b>	<b>Feature methodologies for heterogeneous object realization</b>	<b>225</b>
7.1	Introduction	225
7.2	Literature review	228
7.3	Feature-based design for heterogeneous objects	232
7.3.1	Features	232
7.3.2	Synthesized features for feature-based constructive design	237
7.4	Constructive feature operations and material heterogeneity modelling	240
7.4.1	Constructive feature operations through direct face neighbourhood alteration	240
7.4.2	Material heterogeneity specification for features/objects	245
7.4.3	Example: feature-based design of a prosthesis	248
7.5	Feature-based fabrication in layered manufacturing	250
7.5.1	Staircase interaction in layered manufacturing	250
7.5.2	Staircase interaction-free strategy	251
7.5.3	Implementation	253
7.6	Conclusion and future work	256
7.6.1	Conclusion	256
7.6.2	Future work	257
<b>Chapter 8</b>	<b>CAD modelling and slicing of heterogeneous objects for layered manufacturing</b>	<b>263</b>
8.1	Introduction	263
8.2	Heterogeneous solid modelling	264
8.2.1	A material composition array $M$	264
8.2.2	Material grading function $f(d)$	265
8.2.3	Grading source	266
8.3	Discussion on example cases	270
8.3.1	FGM pipe	270
8.3.2	Watch assembly	272
8.3.3	Cooling fins	273
8.4	Slicing and contours generation for fabricating heterogeneous objects	274
8.4.1	Contours generation	275



	8.4.2	Contours extraction for multiple grading sources	277
8.5		Conclusion	281
<b>Chapter 9</b>		<b>Reverse engineering and rapid prototyping</b>	<b>283</b>
9.1		Introduction	283
9.2		How to obtain point cloud – three-dimensional digitizing	285
	9.2.1	CGI process	287
	9.2.2	Passive acquisition mode	288
	9.2.3	Active acquisition mode	290
	9.2.4	Some commercial digitization systems	298
9.3		From the point cloud to the object – reverse engineering	298
	9.3.1	Surface reconstruction using parametric functions	301
	9.3.2	Surface reconstruction using polyhedron meshing	308
	9.3.3	Rapid fabrication using a point-based segmentation approach	315
9.4		Verification	323
	9.4.1	Introduction	323
	9.4.2	State of the art	325
	9.4.3	The CRAN project	325
	9.4.4	Digitization process planning	325
	9.4.5	Path planning	330
	9.4.6	Application	334
	9.4.7	Conclusion	335
<b>Chapter 10</b>		<b>Virtual prototyping: a step before physical prototyping</b>	<b>341</b>
10.1		Introduction	342
10.2		Virtual prototyping	342
10.3		The .STL format	344
10.4		Previous work	347
10.5		IVECS, the interactive virtual environment for the correction of .STL files	349
	10.5.1	Mode functions	351
	10.5.2	Other efforts	355
10.6		Conclusion	359

<b>Chapter 11</b>	<b>Open issues</b>	<b>363</b>
11.1	Process constraints	363
11.2	Generalization versus specialization	364
11.3	Design constraints (user interface)	365
11.4	Database maintenance and validation	366
11.5	Standards	367
11.6	Data exchange	368
11.7	Design opportunities	368
11.8	Medical issues	370
11.9	Scale	370
11.10	Virtual reality	371
11.11	Conclusion	372

*This page intentionally left blank*

# Introduction

**I Gibson**

Department of Mechanical Engineering, The University of Hong Kong

As time goes on, the complexity of problems and projects has increased to such an extent that it is becoming extremely difficult to develop new technologies in isolation. Universities, research institutes, governments, and research funding bodies recognize this, and the 21st century has started off with an era of collaborative research the likes of which we have never seen before. As a university-based researcher I find it very difficult to keep track of all the new initiatives that are designed to assist collaboration with my peers around the world but, like my colleagues, I find it necessary to look beyond national borders to find out the latest in my chosen research field. One such initiative I came across a couple of years ago was Universitas 21 (U21, which can be found at <http://www.universitas.edu.au>). U21 is a network of universities around the world committed to sharing experiences in teaching and learning. Although this scheme is focused at teaching programmes, I decided to explore the possibility of using it as means of research collaboration. I decided to look for people researching in similar areas to mine within U21 universities. I discovered that there were four universities from the list of 18 who had research groups working in the field of rapid prototyping (RP), and that a common factor between these researchers was that they are all doing at least some work in software systems.

Although not supported by U21 in any way, researchers from these four universities (Michigan, Nottingham, Hong Kong, and Singapore) decided they would investigate ways in which to share research findings and collaborate on a joint project. Dr Jerry Fuh at The National University of Singapore (NUS) was working on the idea of a database that would be used to store information on RP technology and it was considered that maybe this database could be used in different ways by the different groups. Some initial projects were devised and a workshop was scheduled to discuss the results.

This workshop was to be held at The University of Hong Kong. Rather than just limit it to a small group and a single subject, I decided to see if anyone else was interested in participating and extended the scope to a broader area of software systems for RP. After finding there was considerable interest and inviting those who were working in related areas, the result is this book. When writing a collaborative book, I strongly recommend the approach of organizing a workshop. Over 3 days, we worked extremely hard, covering all the material included in this book, with considerable input from all involved. I should also note that we experienced not one but two typhoons during the workshop. While it is not clear as to the significance of their contribution, it certainly focused our efforts – there was nothing else to do.

There are not many texts on RP. Those that exist focus primarily on hardware and technology development. Description of software systems is generally restricted to basic machine operation and model data interchange. The recent development of new technologies and increased commercial interest in RP makes it an appropriate time to produce this book. Software systems research into RP needs to focus on how to deal with new technological developments and assist in proper management of existing technology. By looking at these two problems, the aim of this book is to analyse the state-of-the-art in RP software, discuss the current problems, and determine the directions for new research. The title *Software Solutions for RP* does not mean to imply that this book will provide all the answers. In fact, the authors probably ask more questions than they provide answers. However, there is a definite emphasis on outlining the current work being carried out in the various areas with examples that illustrate how we might arrive at ‘solutions’.

Note also that the book title uses the abbreviated form ‘RP’ rather than rapid prototyping. The historical significance of this is that rapid prototyping is a term that was originally used to define layer-based fabrication processes. The fact that these machines are used for many more purposes than merely prototyping explains the use of the abbreviated term. Just as IBM deals with much more than just business machines, so RP deals with much more than prototyping. This also avoids confusion with other fields that use the same rapid prototyping terminology (e.g. software engineering, chip manufacture, and business development).

The book itself is divided up into a series of chapters that describe different aspects of software systems and attempt to solve the associated problems. The order of these chapters is somewhat arbitrary. I also make no excuses for the inevitable overlaps that different chapters will have. In fact, some chapters will deliberately cover the same material from different perspectives. Since this is an advanced text with a focus on research objectives, it is important to have some basic expectations of the reader. First the reader must know what RP is and have some experience in using the technology and the accompanying software. The reader must also have an understanding of the tools and methods used in software system design. Finally the reader must be prepared to use this book as a base for research rather than expect it to have all the answers.

Chapter 1 starts off by introducing some of the commercial RP operating systems. This is to show that while RP machines have many things in common, the individual characteristics of the processes lead to very specific software features. This leads on to the dilemma presented in the second part of this chapter, which attempts to generalize RP processes so that the model designer can better understand how the final part is going to look.

The second chapter describes the database mentioned earlier in this introduction. This database holds information on different RP machines and the materials they use. Developed at NUS, this database was designed to allow users to view information about RP technology over the Internet. Companies who manufactured RP equipment could also update the database when new technology became available. This chapter describes the database, the fields used, and a web-based software package developed from it.

Benchmarking is an important issue when trying to compare different RP systems. Chapter 3 discusses the various attempts at benchmarking, discussing the information that can be generated from them and how to interpret this information.

Management of RP technology is as important to successful RP use as speed and reliability. Chapter 4 deals with selection of appropriate RP technology. Companies wishing to buy new machines may like to use a software system that can allow them to identify which machine may be most suited to their needs. Companies with RP equipment require software systems that can assist them in preparing quotations and

schedules for themselves and their customers. This chapter discusses the difficulties in generating this information and the theory behind it.

While some of Chapter 1 deals with simulation and process planning, it is felt that there is a need for a specific chapter on this subject. Chapter 5 deals with optimization of RP processes through the use of simulation. Here, rather than just giving the designer feedback on what to expect from RP technology, this chapter tries to automatically determine the best way in which to prepare the model. The specific problem dealt with in this chapter is adaptive slicing.

One particularly interesting development of RP technology is in the field of multiple-material modelling. A number of RP machines are now capable of producing parts with variable material properties. Currently, software representation of such models is only in the early stages of development. Chapters 6, 7, and 8 discuss how to design parts with such properties along with the problems of transferring them to other software packages for analysis and RP. Each chapter has been written by established experts in this field. Since this is such an important topic at this time, I felt it worthwhile to devote a significant amount of the book to it.

Chapters 9 and 10 are devoted to areas related to RP software, but not specifically about RP software. Chapter 9 deals with reverse engineering (RE), discussing the problems related to data capture and RP. The techniques used are somewhat dependent on what needs to be done to the model data before the RP process. A number of examples are shown. One describes how medical data can be transferred quickly and efficiently to the RP machines for part replication. Another example shows that there are many more problems to be faced if the data have to be worked on (smoothing, adding features, etc.) before RP. Chapter 10 deals with the subject of virtual prototyping (VP) and discusses how software can be used in place of RP during some of the conceptualization process. This is a very large topic that is only touched on within this chapter. I think that it is an important issue that should not be overlooked and we have given some recognition to that fact.

There is a final chapter to this book called 'Open Issues'. This was condensed from the various discussions during the book workshop and from the discussions in the other chapters. It is only a snapshot of the many problems that software systems developers are to face in relation

to RP. It is also my own interpretation of these problems and I am sure there are many more and that they can be expressed in many different ways. However, if you are looking for a research direction over the next few years, you could do worse than to look to this chapter. Perhaps also, in the future, we can look to this chapter to see how well it did in defining the software problems in RP.

There are surely many other researchers working on software solutions for RP and there are probably many other projects that do not easily fit into the above categories. For example, we discuss software standards in many of the chapters, but perhaps there should be a separate chapter on the subject of standards. However, I do think that this book paints a good and clear picture of the subject. This is certainly due to the range and expertise of my fellow authors who are undoubtedly amongst the leaders in this subject in the world. All contributed in what is truly a collaborative effort, before, during, and after the workshop. Each chapter was championed by one or more of the team members listed below, but it is important to note that everyone contributed during the whole process. I may have co-ordinated this effort, but there would be no book without the selfless effort of all involved. I hope you benefit from it as much as we did.

#### Team members:

Professor Alain Bernard	IRCCyN Ecole Centrale Nantes, France
Dr Ian Campbell	University of Loughborough, UK
Professor Debashish Dutta	University of Michigan, USA
Professor Georges Fadel	Clemson University, USA
Dr J. Y. H. Fuh	National University of Singapore
Dr Ian Gibson	The University of Hong Kong
Dr Haesong J. Jee	Hong-Ik University, South Korea
Dr Rong-Shine Lin	National Chung Cheng University, Taiwan
Dr H. T. Loh	National University of Singapore
Mr Sebastien Remy	CRAN, University of Nancy, France
Mr Ricky Siu	The University of Hong Kong
Professor S. T. Tan	The University of Hong Kong
Dr Y. S. Wong	National University of Singapore



*This page intentionally left blank*

# Chapter 1

## Advanced pre-processor functionality for layered manufacturing

**H J Jee**

Department of Mechanical Engineering, Hong-Ik University, Seoul,  
South Korea

### ABSTRACT

When considering the use of layered manufacturing (LM), there are many questions a designer might ask in order to handle a computer aided design (CAD) model before he/she practically orders the fabrication. Today, in addition, many people want to do a lot more with CAD files and LM than simply make three-dimensional prints of their CAD files. For example, designers themselves want to use editing/optimization as well as visualization/repair of the STL file to avoid as much physical prototyping as possible since it is expensive to go through a full manufacturing and assembly prototyping. Therefore, the emerging LM industry has created a need for pre-processor software for editing/optimizing/converting STL files into data recognized by an actual LM machine as well as cleaning/repairing ill-defined STL files. This chapter describes general functionality of commercial LM software systems and then introduces advanced pre-processor functionality for LM.

## 1.1 INTRODUCTION

Layered manufacturing (LM, also referred to as rapid prototyping, freeform fabrication, etc.) technologies have an ability of creating a physical part directly from its computer model by adding material on a layer-by-layer basis. To produce a physical model layer by layer in accordance with the geometrical data derived from coherent CAD data, the CAD model is first *sliced* into thin horizontal layers. Each layer is then *processed*, one at a time consecutively, in the LM machine and *bonded* together to form a physical model generically known as a prototype. The major application of LM has been the early verification of product designs and quick production of prototypes for testing. Today, several different types of CAD software exist that can create product geometry for different applications. Once a computer model is generated for a physical object, it needs to be communicated to another computer system for processing. Although it was developed specifically for the stereolithography process, STL polygonal facet representation has become the *de facto* industry standard for the transfer of data to an LM process planning system. CAD model data are now frequently passed to various LM processes using the STL format and, unfortunately, some CAD systems using surface modellers generate bad STL files with geometric flaws. In fact, STL has been notorious for its drawbacks such as redundancy, inaccuracy, and incomplete integrity in the emerging LM community. Most software efforts, not surprisingly, have thus far focused on geometrical verification of the STL CAD model prior to part fabrication (1–3).

Over the years, however, as more and more companies have switched to solid-modelling CAD software systems that almost always produce good STL files, STL fixing has become less of a concern for most users. In addition there are now several very good software products available that will find and fix flaws in STL files. These days, many people want to do a lot more with STL files and LM than simply make three-dimensional prints of their CAD files. For example, designers themselves want to use editing/optimization in addition to visualization/repair of the STL file to avoid as much physical prototyping as possible since it is expensive to go through a full manufacturing and assembly prototyping. Therefore, the emerging LM industry has created a need for pre-processor software for editing/optimizing/converting an STL file into data recognized by an actual LM machine as well as cleaning/repairing ill-defined STL files

(4). This chapter describes general functionality of commercial LM software systems and then introduces advanced pre-processor functionality for LM.

## **1.2 FUNCTIONALITY OF COMMERCIAL LM SOFTWARE**

Basic functionality of current commercial LM software can be generally categorized into three groups: viewing/editing, repair/optimization, and layout/conversion. Table 1.1 shows several examples of the functionality taken from several commercial LM software systems.

### **1.2.1 Viewing/editing**

This functionality provides STL file viewing options such as translate, rotate, and zoom, and it allows users to modify STL files for adding colour information to standard STL files, inscribing text/texture on STL models, and shelling STL parts. It also includes typical editing options such as cutting, adjoining, and scaling CAD models.

### **1.2.2 Repair/optimization**

This functionality is basically for checking STL files for errors (e.g. bad edges or different shells) and also making repairs on this data. It mainly consists of analysing STL files, identifying defects, and either fixing flaws automatically or highlighting them for interactive repair. It also provides functions for reducing the number of triangles in an STL file to optimize file size, smoothing rough areas of a coarse STL file by adding more triangles, and reorienting the STL model to optimize building speed or surface roughness. A separate module for build-time/cost estimation could also support this.

### **1.2.3 Layout/conversion**

This functionality is basically for preparing the build process by placing one or several parts on a three-dimensional platform space (building chamber), generating support structures for parts, and converting the geometry of parts and support structures into data recognized by an actual LM machine. This may also include a tooling option for making moulds from STL files.

**Table 1.1    Functionality of several commercial LM  
software systems**

Main menu		Subsidiary menu	
File		New	
		Open	STL
			STEP
		Save	STL
			STEP
			CLI
			SLC
			SLI
		Close	
		Exit	
Edit		Add	
		Depart	
		Copy	
		Paste	
		Default Position	
		Translation	
		Rotation	
		Scaling	
		Select Part	
		Select All	
View		Redraw	
		Center	
		Zoom	
			All
			In
			Out
			Window
		Rotate	
		Pan	
		View Section	
		Viewpoint	
		Invalid Edge	
		Multi View	
		Grid Step	
		Move View Point	
		View Cross Section By 2	
		View Cross Section By 3	
		Spin about Axis	
		Additional Display Axis	

Table 1.1 continued

Main menu		Subsidiary menu	
Tool		Layout Diagnosis Repair  Slice Virtual Prototyping Network	Fix Normal Merge Vertex Invert Normal Vector Automatic Repair  Listen Connect
Setup		Rendering Viewing View Axis View Platform Window Option	
Help		About Help	
Measure		Distance Point Value	
		Angle Between 3 vertices Radius Arc Edge	
Rendering		Wire-Frame & Shading	
Window		Model Explorer	

1.2.4 Example: three-dimensional-systems stereolithography software

SLA machine software can be divided into two categories: software for off-line programming (3D Lightyear) and software for on-line control of the SLA machine (Buildstation). The functionality of these software systems along with the nature of input and output are described in the following paragraphs.

1.2.4.1 3D Lightyear:

3D Lightyear is a PC-based pre-processing software package for an SLA machine developed by 3D Systems. It works off-line (operated

remotely) and is used to prepare files built on the SLA machine. The software takes STL files of the parts as input and generates a BFF file as output. In the software, the user sets the values of the SLA process parameters (for example, part orientation, layer thickness, and other process-specific parameters). These parameters affect the operation of different components (laser, elevator, and recoater) of the SLA machine.

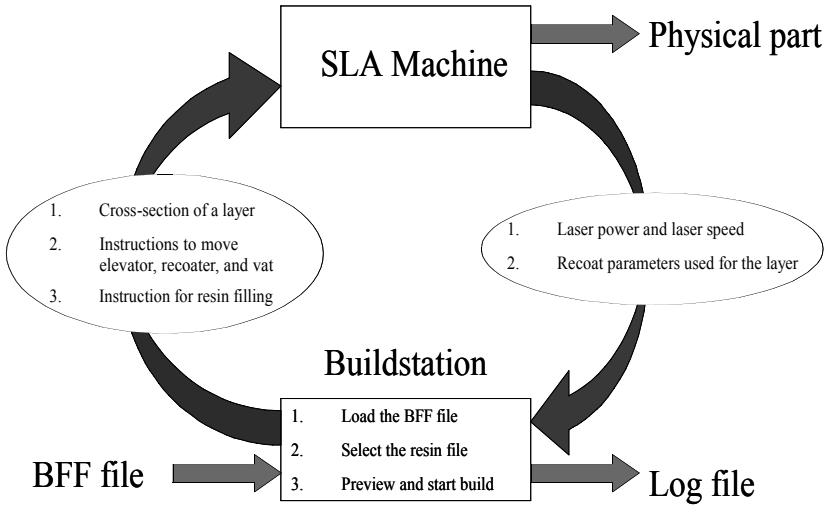
The input and output files for the 3D Lightyear software along with the typical steps used in pre-processing of parts are shown in Fig. 1.1. The first step involves loading the appropriate platform file. A platform file contains information about the type of SLA machine and a set of default values of the process parameters provided by the vendor (3D Systems). The values of all the process parameters are stored in a set of style files. The second step involves loading the STL files of the desired parts. Any number of parts can be loaded as long as they fit in the machine. In the third step, the loaded STL files are verified. This step is helpful in finding and fixing errors in the STL files. Typically, errors in an STL file include inconsistent normal directions (of a triangle) and small cavities in the part. Later, the parts are either automatically or manually positioned in the build platform and support structures are generated for all the parts. Supports are generated automatically, though they can be modified manually if required. In the next step, based on the application, the build and recoat parameters can be modified. Invoking the next step slices the parts into layers and prepares the BFF file. A BFF file contains the cross-section information and the values of process parameters for all the layers in the build.



**Fig. 1.1 Typical steps used in pre-processing parts in 3D Lightyear**





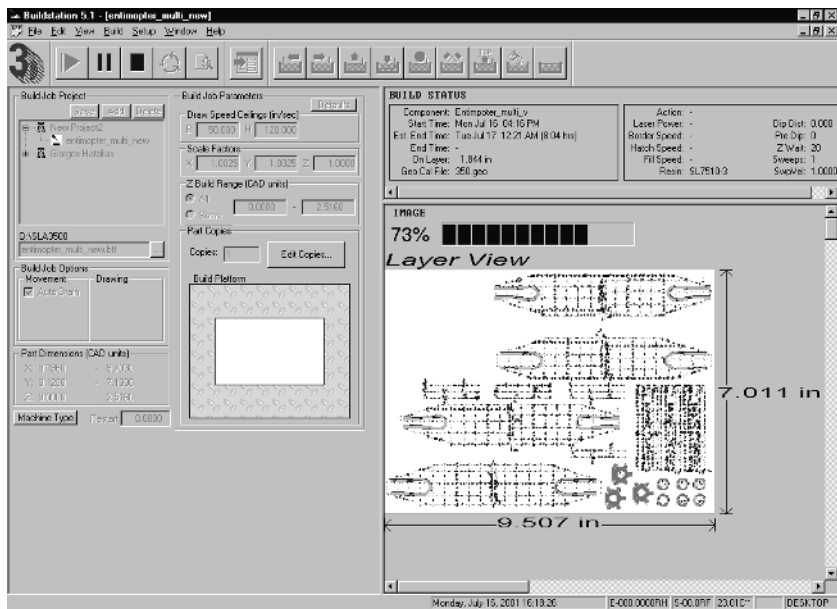


**Fig. 1.3 Information transfer between Buildstation and SLA**

Lightyear on to Buildstation, the appropriate resin file is selected. The resin file contains the laser, resin, and recoating information that is specific to a build. Then, the user can preview the build in layers and if satisfied, start the build. Once the build is started, the Buildstation software transfers the cross-section information and the values of process parameters for the first layer to the SLA machine. This information is used by the SLA machine to build the layer. After building the layer, the values of laser power and the recoat parameters used are transferred to the Buildstation software, which displays these values on the screen to be viewed by the user. Then the information about the second layer is transferred to the SLA machine. This process continues iteratively until the end of the build.

Buildstation is also used to provide instructions to move the elevator, vat, or recoater, or to fill resin in the vat. The laser profile and power can also be viewed.

The secondary functionalities provided by the software include: making multiple copies of the build, providing scale factors to account for the shrinkage, buildtime estimation, and generation of an optional log file with information of the laser power and speed for all the layers of the build. A screenshot of the Buildstation software in action is shown in Fig. 1.4.



**Fig. 1.4 A screenshot of the Buildstation software in action**

### 1.3 ADVANCED FUNCTIONALITY OF LM PRE-PROCESSOR

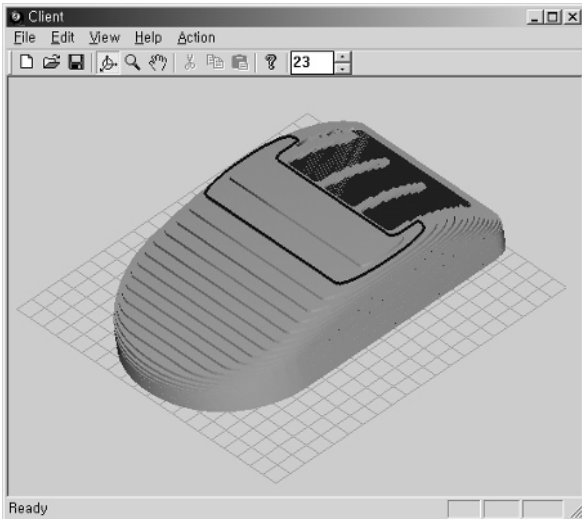
When considering the use of LM, however, there are still many questions a designer might ask when handling an STL model. Moreover, most recent LM processes possess quite different capabilities, and most are not even isotropic. These questions could be answered by a comprehensive design support system providing advanced pre-processor functionality for LM. Our previous approach was to propose visualization and network-based pre-processor as two separate design support modules. Later in this chapter we show how we have improved each separate module and integrated them as a single design support system based on a network.

#### 1.3.1 Intelligent pre-processor based on design visualization

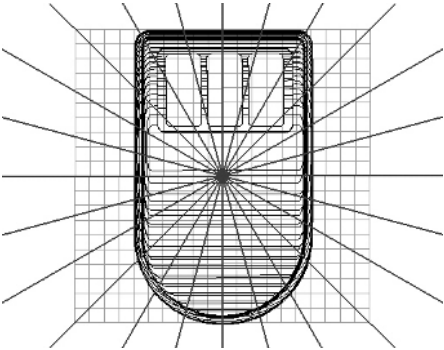
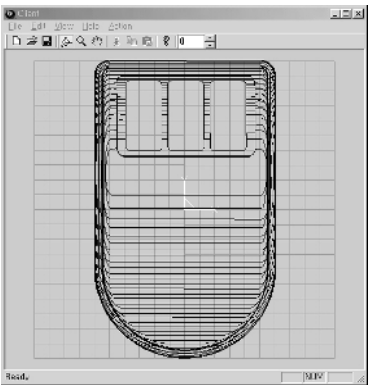
Visualization, in general, is a method of extracting meaningful information from complex data sets through the use of interactive graphics and imaging. It provides processes for seeing and understanding what is normally unseen, thereby enriching existing

scientific methods. Design visualization has become an important part of the computer aided design process. It enables designers to produce realistic images of the final product that can be used as communication aids to other personnel, customers, and suppliers. In many cases, the visualization in the LM community is mainly for verifying surface roughness. When being fabricated by LM apparatus, an object is built by laying down material layers in a gradual, controlled way, which results in the staircase (or laddering) effect on what should actually be smooth surfaces. Traditional geometry-based modellers which display a smooth, shaded surface of the physical model hence provide the designer with no information on the actual surface finish of the object. For example, a sphere fabricated by LM apparatus will look like a stack of discrete, differently sized circular discs rather than a simple smooth-skinned crystal-like ball. Chandru *et al.* (5) have thus suggested voxel-based modelling for evaluating the geometrical effect on the surface of a physical part made by LM technologies, and Jee and Sachs have implemented a visual simulation technique for facilitating surface texture designs (6). Careful examination of the visually simulated model prior to actual fabrication as the authors insisted can hence help minimize unwanted design iterations.

In our previous approach, a visually simulated model can be made by first creating a simple geometry of sliced layer stacks that belong to one part. Each layer is then bonded together, one at a time consecutively, which eventually constructs the whole part (virtual model) as shown in Fig. 1.5. This process is expected to visually simulate the physical process of fabricating a physical part to be made by LM apparatus such as laminated object manufacturing (LOM), stereolithography apparatus (SLA), and fused deposition modelling (FDM). The implemented technique also takes into account accurate geometric attributes of the physical phenomena of the LM machine and hence provides designers with the ability to verify unseen fabrication capability of the existing prototyping machine in the embodiment of their design. In many cases, however, this model still may not be good enough for fully visualizing the surface roughness. It has already been suggested that surface roughness value should be displayed as a particular colour shading on the computer image (7); we therefore add a shaded colour on each layer stack based on the size of the staircase cusp between two adjacent layer stacks in this paper (Fig. 1.6). So the surface roughness value can now be displayed as a particular colour shading together with the visually simulated layered

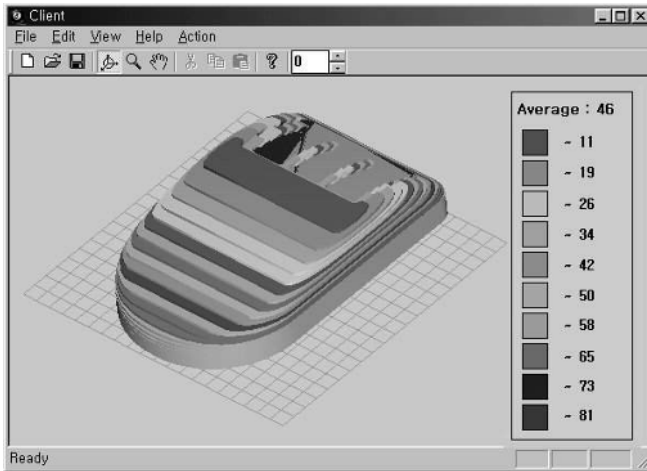


**Fig. 1.5** A simple visually simulated model displaying surface roughness



**Fig. 1.6** Calculation of the size of staircase cusp between two adjacent layers

stacks on the computer image, as shown in Fig. 1.7. This can give the user a clearer visual image of the overall surface roughness and problem areas in the model.



**Fig. 1.7 A visually simulated model displaying surface roughness with shaded colours**

### **1.3.2 Distributed pre-processor based on network**

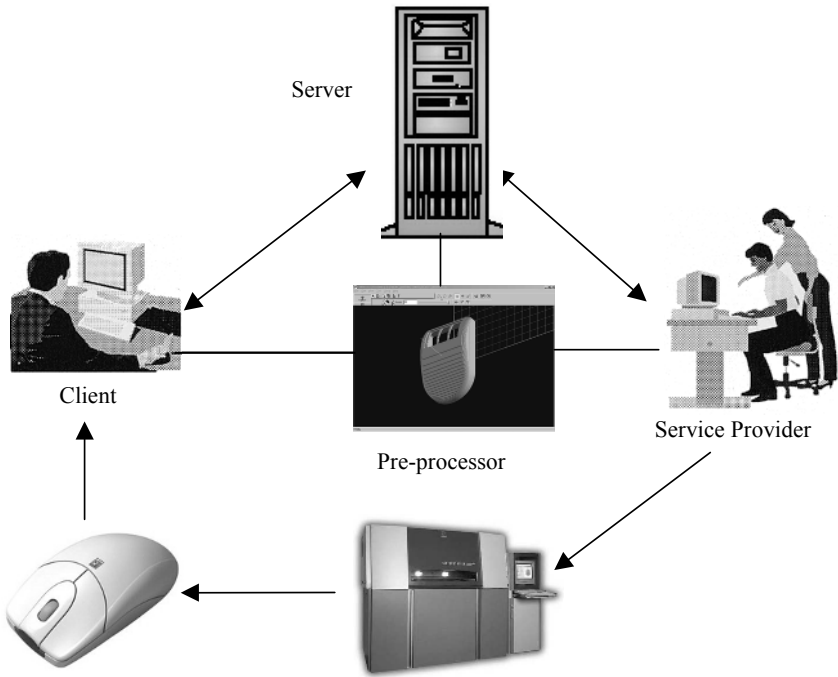
We now live and breathe the networks that we have designed for better communication. Navigating the World Wide Web (WWW) using Internet software, for example, is a highly successful utilization of those communication networks. In the mean time, rapid advances in computing and communication technologies are creating a new approach for product design and manufacturing. It is thought that the lessons learned from the Internet can be applied to LM technologies as well. It is now considered to be important to devise ways of generating physical prototypes through a cheaper and better-amortized process using the Internet, and a group of researchers and developers has already moved towards creating an environment for automated LM capability on the network (8, 9). Called tele-manufacturing facility (TMF), it is creating an automated rapid prototyping (RP) capability on the Internet and undertaking the necessary research and development to ensure that it is viable for engineers and scientists to use over long distances.

When considering the use of LM, in fact, there are many questions a designer might ask. What model orientation should be used, will the model have adequate aesthetic and functional properties, is the STL file suitable for transfer to the LM machine? Designers are now

advised to skip all those preliminary questions and purchase LM services from service providers rather than set up and operate their own equipment. Recently, a number of service providers have been working to establish LM market-places on the Internet in order to save a purchaser time by enabling him/her to submit requests for quotations to multiple service providers simultaneously. They take orders for LM parts through the web pages; designers are supposed to upload their STL files to the company server in electronic form through the Internet by direct upload, ftp file transfer, or as an email attachment; however, many CAD models are still transferred through hand-carried tapes. There is no doubt that the Internet will change the way LM services are delivered. This, however, does not fundamentally change the buying process, and such negotiation still requires a lot of back-and-forth communication between the service purchaser and provider. If the service provider, for example, fixes the STL file but neglects to tell its customer about the rework, the company will probably experience problems later when it tries to use its CAD data to make tooling or to perform the finite element analysis for the customer. Most LM service providers, therefore, need to verify consistency in the STL file in consultation with their customers before the file goes to the LM machine. This step is naturally prone to iteration and delay so that the cost of performing the process becomes virtually unpredictable. Therefore, a simple verification tool for preprocessing STL files on the Internet prior to purchasing LM services could be helpful to designers.

In the previous approach (10), we proposed a framework for a collaborative virtual environment between CAD designers and LM processes on the Internet, which does not push users to purchase expensive commercial software or to hand over their own CAD files to a stranger without knowing what can really happen to their files. Figure 1.8 shows the network-based process configurations for LM pre-processes supported by the proposed method. The proposed pre-processor can be realized by developing a client/server model on the network and directly providing users with a convenient tool, a pre-processor for LM over the Internet, so that it can quickly fill the gap between various CAD systems and distributed LM machines.

In this client/server model the server program continuously listens on one end of the channel, and the client program periodically connects with the server to exchange data. Also, all the distributed LM machines are connected to a server that will manage the process schedule of



**Fig. 1.8 Network-based process configurations for LM pre-processes**

these machines. The server mainly manages a database for handling STL files so that a client can access the server in order to search appropriate STL files. The client can also directly upload and register their own STL files to the server for LM processing. The server will first read and translate the imported STL file into an analytic model for visualization, and the resulting image data will be sent to the client so that it can be displayed on the client terminal. The server also provides clients with the functionality of cleaning/repairing ill-defined CAD files, converting STL files into data recognized by an actual LM machine, and visually simulating fabrication (6) as desired by the LM industry. Each client can therefore order an appropriate slice thickness for the STL file over the network, and an image generated from the slicing can be sent to the client for display on the client terminal. The client can also order a request for visual simulation functionality on the server before the real fabrication of the STL file using LM machines.

The server will then create a virtual model that consists of layer stacks with differently shaded colours and then send image data to the client for display.

## 1.4 DISTRIBUTED INTELLIGENT PRE-PROCESSOR

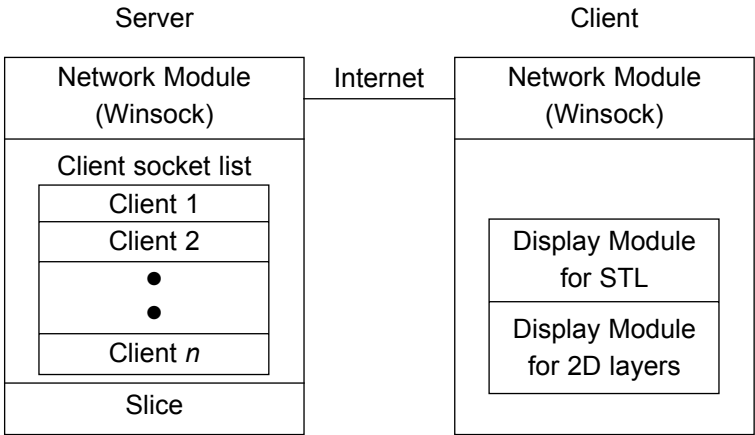
### 1.4.1 System environment

The configuration of the proposed system consists of three main parts: server, client, and the network in between, as shown in Fig. 1.9. The network provides all necessary logic for supporting the server–client interface and controls the network connection for system development. In constructing the LM pre-processor on the network, we proposed Window socket (*Winsock*) programming for server/client networking, and Visual C++ and OpenGL in a Windows operating system environment for three-dimensional geometric modelling. For Winsock programming, CSocket class inherited from CAsyncSocket in MFC library Version 6.0 is used. Winsock is used to exchange packets of data between the server end and the client end across the network. Winsock is the lowest level Windows Application Programming Interface (API) for Transmission Control Protocol/Internet Protocol (TCP/IP). Both the Internet server programs and client programs can be written using the Winsock API. Most of the Winsock calls are made from worker threads so that the program's main thread can carry on with the user interface. One of the challenges in socket programming, however, is to prepare for the possibility of a broken network during data exchange. It is thus sometimes required to identify the source of the problem; whether it is a network problem, a software problem, or a server/client problem.

The proposed client/server networking in this paper is also possible in the local network as well as the Internet using IP address; JAVA applet and JAVA3D can likewise be used for developing the proposed system. The system using JAVA web browser may have some advantages: it will be more intuitive to the users and no complex network construction process is required. The loading speed of JAVA applet, however, is still not fast enough for current network applications. In addition, JAVA3D library based on OpenGL has difficulty in providing images with good graphic quality and quick execution time. Compared with the JAVA applet, the advantages of our system can be summarized as follows:



- It can reduce the execution time since it is based on the client application program rather than the web browser.
- It can provide three-dimensional images with better graphic quality using Windows-based OpenGL.
- It can make fast data transmission between the server and client using Winsock programming.



**Fig. 1.9 System configuration for the network-based pre-processor**

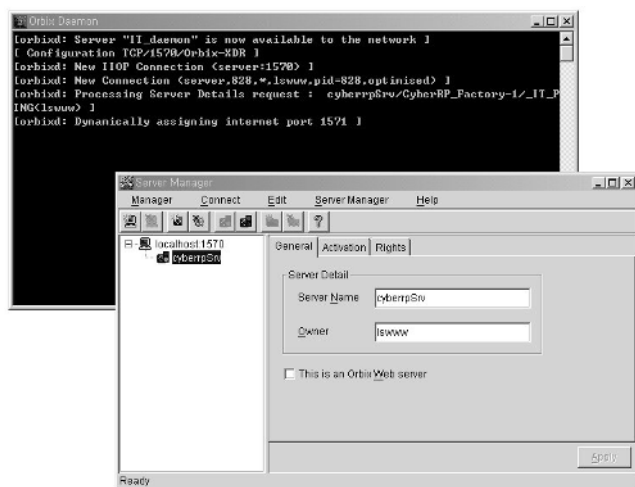
**1.4.2 System implementation**

The entire Internet namespace is generally organized into domains, starting with an unnamed root domain. Below the root is a series of top-level domains such as com, edu, gov, and org. When it comes to the server end first, suppose a company named CyberRP has a host computer connected to the Internet for WWW service. By convention, this host computer is named `www.CyberRP.com`, and CyberRP designates a host computer as its name server. The name server for the .com domain has a database entry for the CyberRP domain, and that entry contains the name and IP address of the CyberRP name server. Now for the client side. A user types `http://www.CyberRP.com` in the browser. The browser must then resolve `www.CyberRP.com` into an IP address, so it uses TCP/IP to send a Domain Name System (DNS) query to the default gateway IP address for which TCP/IP is configured. Assuming that client and server computers were connected to the Internet, it is possible to run exactly the same client and server

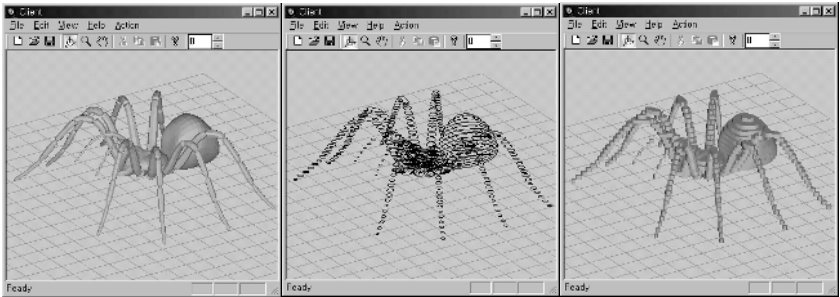
software on a local intranet. An intranet is often implemented on a company's local area network (LAN) and is used for distributed applications. Users see the familiar browser interface at their client computers, and server computers supply simple web-like pages or do complex data processing in response to the user input as shown in Fig. 1.10. Figure 1.11 shows a viewing image of an example STL file (left), its two-dimensional sliced wired layers image (middle), and its visually simulated LM part image (right) displayed over the network. Figure 1.12 shows a visually simulated model displaying surface roughness together with shaded colours over the network. If the resulting image looks satisfactory, users can assign a working order for LM fabrication to the server. The server can then search an appropriate LM service bureau and pass the order in order to execute the fabrication.

## 1.5 CONCLUSIONS

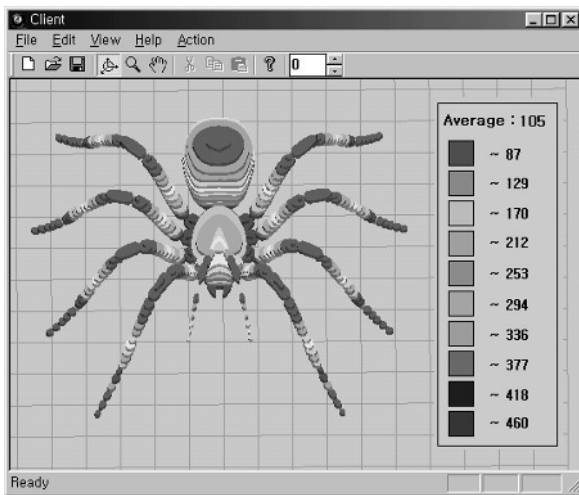
This paper proposes an advanced LM pre-processor on the Internet with the functionality of cleaning/repairing ill-defined CAD files, visualizing/optimizing the LM part for decision support, and converting STL files into data recognized by an actual LM machine. A client/server model on the network is developed to realize the pre-processor, and a software tool with the functionality for the pre-



**Fig. 1.10 The server manager controlling the network connection with clients**



**Fig. 1.11** Different viewing modes of an example STL file



**Fig. 1.12** An image displaying surface roughness with shaded colours

processor has been implemented on this model. The method provides for an environment for automated LM capability by developing a framework for a collaborative virtual environment between various CAD software and different/distributed LM processes on the Internet. Using this method, smooth data transfer from CAD to LM is possible over the network; this enables users to generate physical prototypes through a cheaper and better-amortized process using the Internet.

The biggest potential barrier in realizing this method, however, will be the security of design data; designers often tend to be paranoid about

protecting their newest creations. Even with the most sophisticated encryption methods to ensure that files are not poached by hackers, designers will be reluctant to upload their files on to the Internet without iron-clad non-disclosure agreements from service providers. Since it is possible to run exactly the same client and server software on a local intranet, however, an intranet is often implemented on a company's local area network (LAN) and is used for distributed applications. Additional development of direct tele-manufacturing functionality using tele-control of LM machines is needed in this pre-processing environment. In addition, as it becomes more complex to work with LM, tooling design is also indispensable for soft and hard tooling industries.

## REFERENCES

- 1 Bohn, J. H. and Wozny, M. J. (1992) Automatic CAD-model repair: shellclosure. In Proceedings of the SFF Symposium, Department of Mechanical Engineering, University of Texas at Austin, pp. 86–94.
- 2 Morvan, S. M. and Fadel, G. M. (1996) IVECS Interactively correcting .STL files in a virtual environment. In Proceedings of the SFF Symposium, Department of Mechanical Engineering, University of Texas at Austin, pp. 491–498.
- 3 Krause, F.-L., Stiel, C. and Luddemann, J. (1997) Processing of CAD-data – conversion, verification and repair. In Proceedings of 4th ACM Symposium on Solid Modeling, Atlanta, Georgia, pp. 248–254.
- 4 Barequet, G. and Kaplan, Y. (1998) A data front-end for layered manufacturing. *Computer-Aided Design*, **30**(4), 231–243.
- 5 Chandru, V., Manohar, S. and Edmond Prakash, C. (1995) Voxel-based modeling for layered manufacturing. *IEEE Computer Graphics and Applications*, November, 42–47.
- 6 Jee, H. and Sachs, E. (2000) A visual simulation technique for 3D printing. *Advances in Engineering Software*, **31**(2), 97–106.
- 7 Campbell, R. I., Jee, H. and Lee, H. (2000) Visualization tools for design support in SFF. In Proceedings of the SFF Symposium, Department of Mechanical Engineering, University of Texas at Austin, pp. 437–444.
- 8 Bailey, J. M. (1995) Tele-manufacturing: rapid prototyping on the Internet. *IEEE Computer Graphics and Applications*, **15**(6), 20–26.

- 9 Luo, C. R., Lee, Z. W., Chou, H. J., and Leong, T.H. (1999) Tele-control of rapid prototyping machine via Internet for automated tele-manufacturing. In IEEE International Conference on *Robotics and Automation*, **3**, 2203–2208.
- 10 Jee, H. and Lee, S. (2000) An RP preprocessor over the Internet. In Proceedings of the 9th European Conference on *Rapid Prototyping and Manufacturing*, Athens, Greece, pp. 65–77.

# Chapter 2

## A web-based database system for RP machines, processes, and materials selection

**J Y H Fuh, H T Loh, Y S Wong, D P Shi, M Mahesh, and T S Chong**  
Department of Mechanical Engineering, National University of  
Singapore, Singapore

### ABSTRACT

In this chapter, the development of a web-based rapid prototyping (RP) database for machine, processes, and materials selection is described. The RP website allows users to be kept informed of the latest developments in the RP arena by soliciting RP vendors to update their information on the website. At the same time, this website can be incorporated with the proposed ‘web-based RP decision support system’.

### 2.1 INTRODUCTION

With the rapid development of rapid prototyping and manufacturing (RP&M) technologies, RP prototypes are being widely used in many applications for new products, including conceptual design, functional prototypes, and patterns for tooling. Owing to the variety in RP&M technologies and processes, which have very different characteristics and thereby result in prototypes with quite different properties, RP&M users or customers usually need to make several planning decisions according to the application’s requirements. A ‘web-based RP decision support system (RPDSS)’ has been proposed by the research collaborating team from the National University of Singapore (NUS),

The University of Hong Kong, Loughborough University, and the University of Michigan. The system provides a platform mounted on the NUS server. RP users or customers can access the system via Internet/Intranet to make a suitable decision according to the application's requirements. The overall architecture and main modules of the proposed RPDSS system are shown in Fig. 2.1.

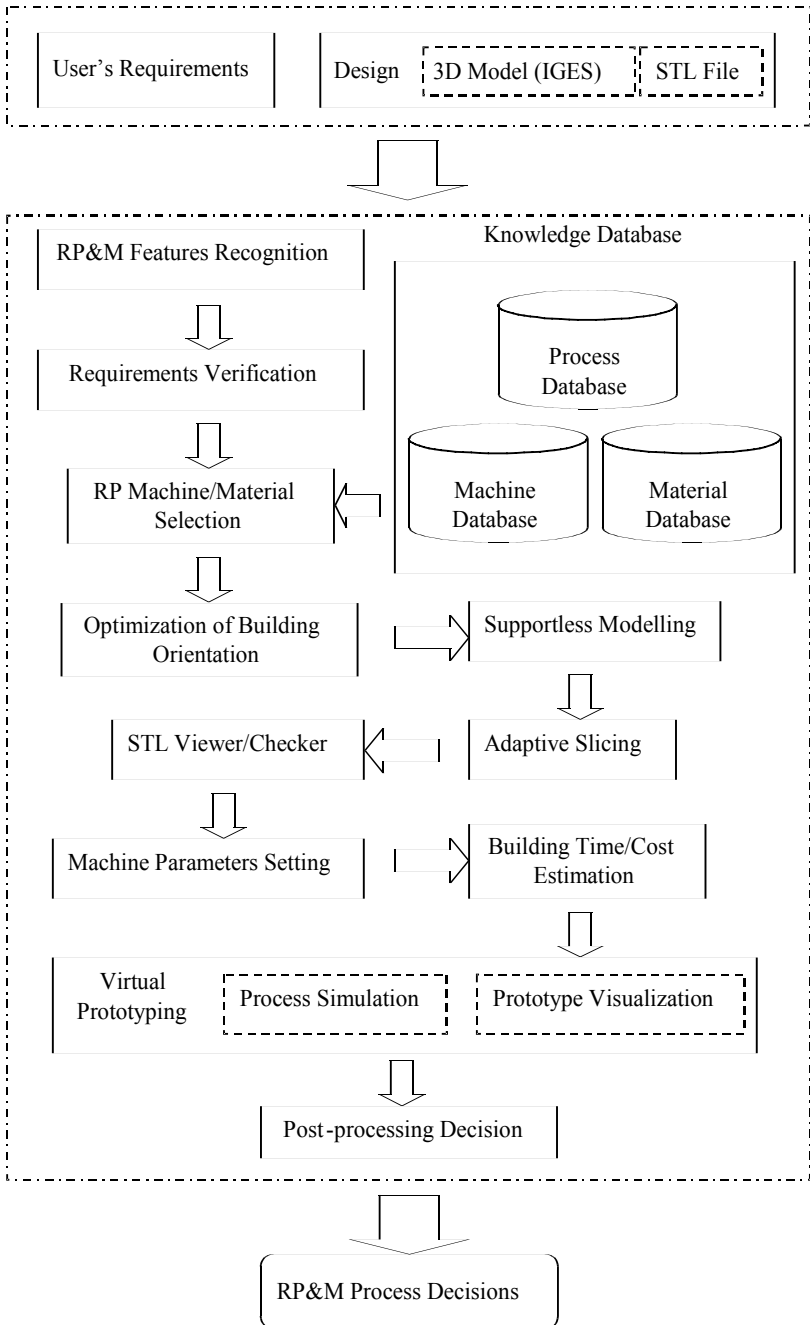
In the past, before the emergence of new manufacturing technologies grouped under RP, manufacturing a part of irregular shape could take weeks, sometimes months. However, with rapid prototyping, the manufacturing time of such complex parts can be reduced to hours. The working principle of this technology is that parts are built in a layer-by-layer manner. They can be classified into three different types of RP systems: liquid based such as stereolithography apparatus (SLA); solid based such as laminated object manufacturing (LOM) and fused deposition modelling (FDM); and powder based such as selective laser sintering (SLS). Therefore, to aid RP users when they have doubts or queries, there must be a place where they can go for consultation or advice, a place for them to make comparisons or benchmark the different RP equipment manufactured by different companies but having similar functions. Information presented to buyers must be comprehensive enough for them to decide which process to use and which equipment to buy.

### **2.1.1 RP website**

In this section, the feasibility of hosting a website to solve the problem mentioned in the previous section is discussed. The world has become smaller with the rapid advancement of information technology (IT). IT has such a great impact on people, the most influential of which is the Internet. It has become part of daily routine that people in developed countries cannot live without. Thus, creating a website to showcase different kinds of RP equipment to potential RP buyers is the most appropriate, efficient, and cheapest way. The advantages of having a RP website are outlined below.

#### ***2.1.1.1 Time-saving***

It takes less time to store new RP product information to the website's database than mailing out new RP product brochures to potential RP buyers.



**Fig. 2.1 Main modules of RPDSS**



#### **2.1.1.2 *Unconstrained boundary***

Anyone from any location can access the website as long as they are connected to the Internet.

#### **2.1.1.3 *Real time***

RP information stored in the database can be seen instantly from the user's browser. Therefore, RP buyers are always updated with the latest information on RP technology.

#### **2.1.1.4 *Standardized information***

As the website structure presents the same interface to every RP vendor, information stored will be standardized. Hence, anyone who accesses the data can make a fair comparison of the different RP equipment.

#### **2.1.1.5 *Ease of tracking***

The popularity of the website can be judged by programming the site in such a way that the number of people accessing the site is monitored. It can also keep track of the date and time at which RP information is stored and updated by individual companies.

#### **2.1.1.6 *Cost effectiveness***

It will be much more cost effective, even with the software and hardware required to create the website, than hiring more staff just for the preparation of mailing out new RP product leaflets.

### **2.1.2 Objectives**

The main objective of this development is to design a website for RP vendors to showcase their company's product and for RP buyers, or those interested in RP, to update themselves on the newest RP products that have been launched on the commercial market. The website must also be able to support the system which NUS together with three other universities (University of Hong Kong, University of Michigan, and Loughborough University) are working towards, a 'web-based RP decision support system'.

A 'web-based RP decision support system' acts as an RP consultant that gives advice to people over the net except that this 'expert' is remotely located. Users have to send over the acceptable drawing file

format and this ‘consultant’ will analyse it and give advice on the drawing such as:

- Geometry like the minimal wall thickness, through and blind holes, etc.
- Material and mechanical properties like the tensile strength, surface hardness, etc.
- Prototypes’ quality on the dimensional accuracy in  $X$ ,  $Y$ , or  $Z$  orientation, etc.

The RP consultant will then run through the RP database to retrieve the most suitable RP equipment for the completion of the drawing. However, before this ‘RP consultant’ can do its job, a detailed study must be carried out on the benchmarking of equipments and processes available.

### **2.1.3 Development backgrounds**

The concept of putting information on a website and storing it in a database system is not totally new. There are several web-based database systems reported (1–7) for various applications. Some of the useful websites for RP can also be found from references (8–14). A website can be seen as an effective means to transmit information and hence, with its growing popularity, manufacturing companies store their technical product drawings in the hard-copy under shelf or as a soft-copy in an isolated computer terminal. Only people from a particular company can access the drawings of that company. However, in recent years, more and more database systems are going on-line. Thus, companies begin to set up servers and store data in database systems so as to ease the accessing process for people who are interested in getting information about the company’s product. The rapid prototyping database is conceived in such a background.

## **2.2 METHODOLOGY**

After the selection of the web design tool and database system for the website, the development work can then go into the website design and the construction of tables in the database. It is important to design a user-friendly website so that users will visit again. It is also vital to construct the tables in the database carefully for future expansion purposes. In this section, the planning involved in the design of a website, construction of the tables, and notification of the RP website to vendors will be shown.

The development process of this website can be divided into two main phases. They are:

- Website design
- RP data collection

### 2.2.1 RP website design

In the previous section, the selection of the web design tool and database system is mentioned. Thus in this section, the more detailed information on the development of software and hardware will be discussed. The software followed by the hardware aspects for website design are described.

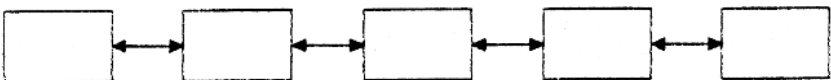
#### 2.2.1.1 Software selection

Under the software section, readers will get to know how the website structure is chosen and how the construction of tables and queries is done. Therefore, a good understanding of how RP works is necessary to design a good RP website. The website must be user friendly and all the information captured must be essential to give people comprehensive information of each RP product.

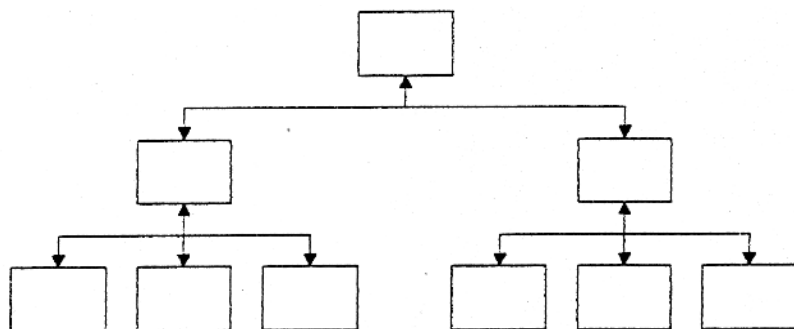
#### 2.2.1.2 Determining the website structure

The website can be structured into four fundamental structures. They are:

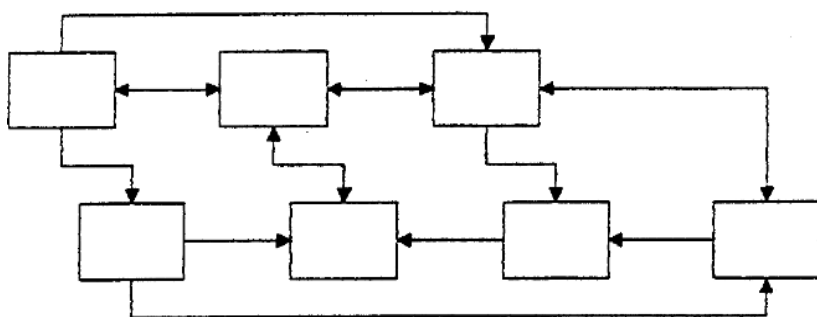
- *Linear* – a simple sequential structure which users are expected to follow rigidly. This can be seen in Fig. 2.2.
- *Hierarchical* – a logical tree structure where users have slightly more freedom to navigate along its branches. This can be seen in Fig. 2.3.
- *Non-linear* – an unrestricted navigation. Users are not restricted to predefined routes. This is shown in Fig. 2.4.
- *Composite* – users are allowed to navigate freely with occasional restriction to follow a linear structure through critical information. This can be seen in Fig. 2.5.



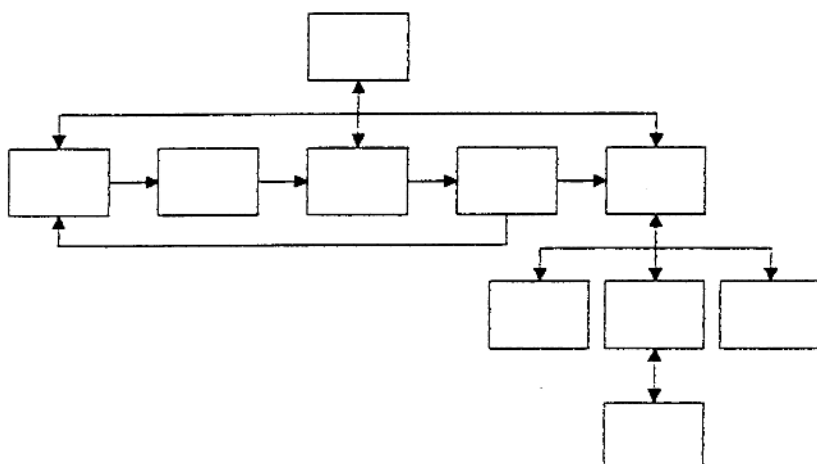
**Fig. 2.2 Linear structure**



**Fig. 2.3 Hierarchical structure**



**Fig. 2.4 Non-linear structure**



**Fig. 2.5 Composite structure**

The design structure chosen for the website is the composite structure. In a composite structure, there is a combination of freedom and restriction to users. Users are free to navigate in the general information level but when it comes to the important information, they will follow through a linear structure that can be seen in Fig. 2.2. The reasons for not choosing the other alternatives are as follows:

- Linear and hierarchical structures restrict the users too much which will make users feel that the website is inflexible.
- Non-linear structure gives too much freedom to users which might be confusing for new visitors.

### ***2.2.1.3 Construction of tables***

After finalizing the website structure, the construction of tables in the database comes next. The tables constructed are crucial, as this will affect future amendments to the database. The tables are designed so that there is a table for each piece of RP information. For example, there is a table for RP machines, a table for RP vendors, and a table for RP material, etc. There are a total of 12 tables (Tables 2.1 to 2.12) constructed for this development. A brief description of each table is shown below.

*RP machine table*

Purpose: to store all characteristics of each RP machine.

**Table 2.1 Machine table**

Field Name	Data Type	Description
MachineID	Number (integer)	Machine ID number (Pri. Key)
MachineName	Text (30)	Machine name
MachineTypeID	Number (integer)	Machine type ID number (F. Key)
MaxPower	Text (20)	Maximum laser power (W)
X_BuildingSize	Number (single)	Max building dimension in X (mm)
Y_BuildingSize	Number (single)	Max building dimension in Y (mm)
Z_BuildingSize	Number (single)	Max building dimension in Z (mm)
Z_Accuracy	Number (single)	Accuracy in Z direction (mm)
X-Y_Accuracy	Number (single)	Accuracy in X-Y direction (mm)
MiniWall	Number (single)	Mini wall dimension (mm)
MiniHole	Number (single)	Mini hole diameter (mm)
MaxSpeed	Number (single)	Maximum beam speed (m/s)
MachineDimension	Text (50)	Machine dimension (mm x mm x mm)
Weight	Number (single)	Machine weight (kg)
DataFormat	Text (50)	CAD data format
MinThickness	Number (single)	Minimum slice thickness (mm)
MaxThickness	Number (single)	Maximum slice thickness (mm)
ElectricalSupply	Text (50)	Electrical supply of the machine
VendorID	Number (integer)	Vendor ID number (F. Key)
Price	Currency	Price of one unit (US\$)
Registered_Date	Date/Time	Info on RP product register date
Registered_Time	Date/Time	Info on RP product register time
Updated_Date	Date/Time	Info on RP product updated date
Updated_Time	Date/Time	Info on RP product updated time

*RP materials table*

Purpose: to store RP materials information.

**Table 2.2 Materials table**

Field Name	Data Type	Description
InitialName	Text (20)	Material initial name
Descriptions	Memo	Description about the material
Price	Currency	Price of 1 kg
SpecificGravity	Number (single)	Specific gravity (g/cm <sup>3</sup> )
Density	Number (single)	Powder density (g/cm <sup>3</sup> )
ParticleSize	Text (20)	Particle size range (μm)
Colour	Text (20)	Colour of material
SpecificHeat	Number (single)	Specific heat, Cp (J/kg K)
ThermalCond	Number (single)	Thermal conductivity, k (W/m K)
MeltingTemp	Number (single)	Melting temperature, Tm (°C)
GlassTransTemp	Number (single)	Glass transition temperature, Tg (°C)
Viscosity	Number (single)	Viscosity at 30°C (Pa s)
Shrinkage	Number (single)	Shrinkage value (%)
HeatTransCoeff	Number (single)	Heat transfer coefficient (W/m <sup>2</sup> K)
VendorID	Number (integer)	Vendor ID number (F. Key)
Registered_Date	Date/Time	Info on RP product register date
Registered_Time	Date/Time	Info on RP product register time
Updated_Date	Date/Time	Info on RP product updated date
Updated_Time	Date/Time	Info on RP product updated time

*RP applications table*

Purpose: to store all applications information.

**Table 2.3 Applications table**

Field Name	Data Type	Description
VendorID	Number (integer)	Vendor ID number (F. Key)
ApplicationID	Number (integer)	Application ID number (Pri. Key)
ApplicationName	Text (30)	Application name
Description	Memo	Description about the application
Registered_Date	Date/Time	Info on RP product register date
Registered_Time	Date/Time	Info on RP product register time
Updated_Date	Date/Time	Info on RP product updated date
Updated_Time	Date/Time	Info on RP product updated time

*RP vendors table*

Purpose: to store RP vendors information.

**Table 2.4 Vendors table**

Field Name	Data Type	Description
VendorID	Number (integer)	Vendor ID number (Pri. Key)
Name	Text (30)	Vendor name
LogName	Text (50)	Log-in Name
Address	Text (200)	Vendor address
City	Text (20)	Vendor city
State	Text (20)	Vendor state
PostalCode	Text (20)	Vendor postal code
Country	Text (20)	Vendor country
Email	Text (50)	Company Email
WebSite	Text (40)	Vendor Internet website
TelephoneNumber	Text (20)	Vendor telephone number
FaxNumber	Text (20)	Vendor fax number
Registered_Date	Date/Time	Info on RP Vendor register date
Registered_Time	Date/Time	Info on RP Vendor register time
Updated_Date	Date/Time	Info on RP Vendor updated date
Updated_Time	Date/Time	Info on RP Vendor updated time

*RP machine type table*

Purpose: to store machine type information.

**Table 2.5 Machine type table**

Field Name	Data Type	Description
MachineTypeID	Number (integer)	Machine type ID number
MachineType	Text (50)	Machine type



*RP mechanical properties table*

Purpose: to store important mechanical properties of prototypes.

**Table 2.6 Mechanical properties table**

Field Name	Data Type	Description
MechanicalID	Number (integer)	Mechanical ID number (Pri. Key)
MaterialID	Number (integer)	Material ID number (F. Key)
YoungModulus	Number (single)	Young's modulus (MPa)
TensileStrength	Number (single)	Tensile strength at yield (MPa)
ImpactStrength	Number (single)	Impact strength (MPa)
Hardness	Text (50)	Surface hardness
Elongation	Number (single)	Tensile elongation at break (%)

*RP surface qualities table*

Purpose: to store some surface qualities of prototypes.

**Table 2.7 Surface qualities table**

Field Name	Data Type	Description
PrototypeID	Auto Number (long integer)	Prototype quality ID number (Pri. Key)
MechanicalID	Number (integer)	Mechanical ID number (F. Key)
PlanSurfUpRough	Number (single)	Planar surface in upward ( $\mu\text{m}$ )
PlanSurfDoRough	Number (single)	Planar surface in downward ( $\mu\text{m}$ )
PlanSurfSiRough	Number (single)	Planar surface in sideward ( $\mu\text{m}$ )
SlopeSurfUpRough	Number (single)	Slope surface in upward ( $\mu\text{m}$ )
SlopeSurfDoRough	Number (single)	Slope surface in downward ( $\mu\text{m}$ )
FreeSurfUpRough	Number (single)	Freeform surface in upward ( $\mu\text{m}$ )
FreeSurfDoRough	Number (single)	Freeform surface in downward ( $\mu\text{m}$ )
DiAccuracy_XY	Number (single)	Dimensional accuracy in X–Y-plane
DiAccuracy_Z	Number (single)	Dimensional accuracy in Z-axis

*RP tolerance table*

Purpose: to store the required tolerance values.

**Table 2.8 Tolerance table**

Field Name	Data Type	Description
ToleranceID	Number (integer)	Feature ID number (Pri. Key)
Perpendicularity	Number (single)	Perpendicularity value ( $\mu\text{m}$ )
Flatness	Number (single)	Flatness value ( $\mu\text{m}$ )
Parallelism	Number (single)	Parallelism value ( $\mu\text{m}$ )
PosTolerance	Number (single)	Positioning tolerance value ( $\mu\text{m}$ )
DiaTolerance	Number (single)	Diameter tolerance value ( $\mu\text{m}$ )
Cylindricity	Number (single)	Cylindricity value ( $\mu\text{m}$ )
Circularity	Number (single)	Circularity value ( $\mu\text{m}$ )
DepthTolerance	Number (single)	Depth tolerance ( $\mu\text{m}$ )
ConicalTaper	Number (single)	Conical taper value ( $\mu\text{m}$ )
Squareness	Number (single)	Squareness value ( $\mu\text{m}$ )
SurfaceProfile	Number (single)	Surface profile value ( $\mu\text{m}$ )

*RP fabrication time table*

Purpose: to store fabrication time factors.

**Table 2.9 Fabrication time table**

Field Name	Data Type	Description
FabTimeID	Number (integer)	Fabrication time ID number (Pri. Key)
LayerDelay	Number (single)	Delay time between layers (s)
Elevating	Number (single)	Elevating time (s)
Coating	Number (single)	Coating time (s)
Preparation	Number (single)	Data preparation time (h)
Finishing	Number (single)	Finishing time (h)

*RP fabrication cost table*

Purpose: to store fabrication cost factors.

**Table 2.10 Fabrication cost table**

Field Name	Data Type	Description
FabCostID	Number (integer)	Fabrication cost ID number (Pri. Key)
Salary	Number (single)	Salary of the operator (US\$/h)
MachineRun	Number (single)	Machine running cost (US\$/h)
Computation	Number (single)	Computation cost (US\$/h)
Cleaning	Number (single)	Cleaning cost (US\$/h)
Postcuring	Number (single)	Post curing cost (US\$/h)
Finishing	Number (single)	Finishing cost (US\$/h)

*RP material type table*

Purpose: to store material type information.

**Table 2.11 Material type table**

Field Name	Data Type	Description
MaterialTypeID	Number (integer)	Material type ID number
MaterialType	Text (50)	Material type

*Finishing method table*

Purpose: to store finishing method information.

**Table 2.12 Finishing method table**

Field Name	Data Type	Description
FinishingMethodID	Number (integer)	Finishing method ID number
FinishingMethod	Text (50)	Finishing method

**2.2.1.4 Construction of queries**

After constructing the tables in the database, queries can then be formed. Queries are nothing more than linking some tables together as one. The purpose is to check the relationship between each table. This makes data retrieval easier as the program does not need to open a few tables at one time to retrieve information from each table. It can be retrieved from the query section of the database system. The

following is a brief description of each query used for the website.

Material Type Query (qryMaterial)

Purpose: to query vendors and material specifications.

Source: tblVendor, tblMaterial

Field: all fields of tblMaterial, all fields of tblVendor, VendorAddress.

#### **2.2.1.5 Hardware section**

With the software section completed, the hardware section is another important area to be discussed and analysed. Hardware refers to the physical items. Thus, in order to host a website, the following hardware is needed:

- A server with a pre-installed database system such as Microsoft Access 2000.
- A fixed internet protocol (IP) with a registered domain name. (Examples of domain names are *www.nus.edu.sg* and *www.microsoft.com*. This is to ensure that when people type in the domain names in the URL, they will be brought to the correct venue.)

#### **2.2.2 RP data collections**

With the software and hardware parts completed, the next thing is to acquire the RP information and store it in the project database. There are many ways to acquire RP information; examples are given below:

- Browsing through the Internet to search for RP product information.
- Paying a visit to an individual RP company to get product brochures.
- Calling up an individual RP company to inquire about product information.
- Getting an individual RP company to provide their own product information.

The fourth way is the best way to acquire RP information. It saves time in inserting data into the database because the companies know their product best. In order to achieve that, an invitation email is sent out to every RP company that has an email address. The content of the invitation email will inform them of the launching of an RP website and give the URL to this website. They are cordially invited to register their company and then update their RP information in the database.

### 2.2.3 Copyright issues

There should not be any problems with the copyright issues if any dispute arises. The invitation email sent out to all RP companies clearly asked for permission to showcase their RP products. In case any company should turn down the offer, all data pertaining to that company and the RP product that the company is carrying will be deleted to avoid intellectual property rights (IPR) issues.

### 2.2.4 Selection of web-based design tools

The problem with static HTML is that it is a one-way communication. There is no way to send information back to a web server. Active server page (ASP) is an abbreviation for application service provider. ASP (13,14) is an HTML page that includes one or more scripts (small embedded programs) that are processed on a Microsoft web server before the page is sent to the user. It is a server-side scripting technology for building web pages that are both dynamic and interactive. Typically, the web server uses input received from the user to access data from a database and then builds or customizes the page before sending it back to the requestor. Created ASP files can also include scripts written in VBScript, JScript, and ActiveX Data Objects program statements. An ASP file can be easily differentiated from the rest by the '.asp' file suffix. Microsoft recommends the use of the server-side ASP rather than a client-side script wherever possible, because the server-side script will result in an easily displayable HTML page. Client-side scripts (for example, with JavaScript) may not work as intended on older browsers.

Hence, the advantages of using ASP are as follows:

- It solves all the problems associated with CGI and server APIs.
- It is as efficient as internet server application program interface (ISAPI) applications.
- It makes web pages easier to maintain and update in the future.
- ASP is a lot simpler to learn and ASP codes can be written in the HTML page itself.

The HTML tags and the codes are arranged side by side. Codes are written in a simple scripting language that is easy to understand. After saving the page to the website, it is ready to be used. No compiling and complex interfacing are needed.

## 2.2.5 Choosing a database system

A database is a system which specifies data by allocating data types, structures, and constraints for the data. Some examples include text of 50 characters or an integer number. It constructs the data by storing data under management and control. Some examples include a table created for gender and a table created for student names. It also manipulates the data by giving query to update, generate reports, etc. In general, a database system is a tool to help one organize and manage data (and make it available as information). The key goal of database systems is to aid in decision-making processes.

Nowadays, there are a lot of database systems available in the commercial market such as Oracle or SQL Server. These are full database servers. A company will go for one of these database systems if they have a large amount of information to be kept because these systems' engines are superior in terms of speed and have multi-user capabilities. In the case of this development, Microsoft Access 2000 is chosen instead of Oracle or SQL Server due to the ease of use. The main reasons are as follows:

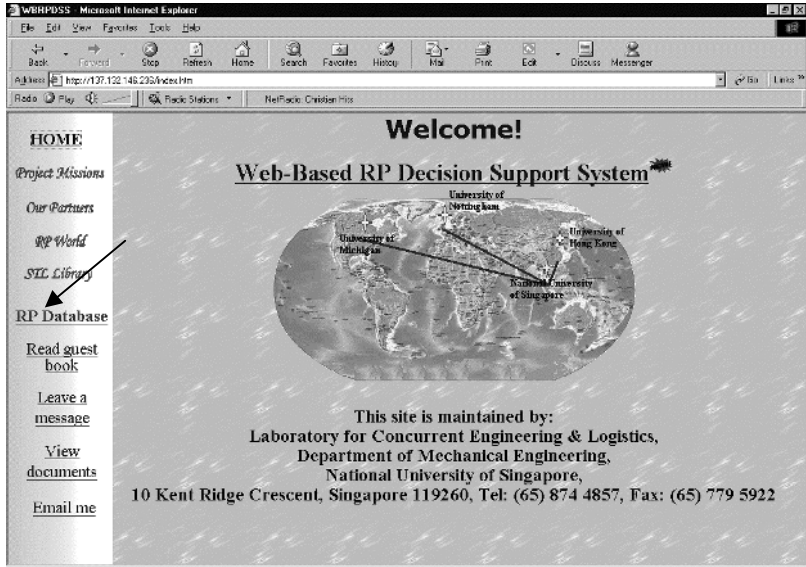
- Microsoft Access has millions of users and therefore there are a lot of add-ons and other third-party software available.
- There are a lot of books and articles written and numerous sites on the Web on Microsoft Access. Hence, if there are any problems or doubts with Microsoft Access, solutions can easily be found.
- Microsoft Access can be used by almost anyone of any level. Beginners can get to grips with it very easily using the wizards and the easy-to-understand interface while developers can push it to its limits and do some extraordinary things with it. It is a user-friendly system.
- In terms of cost, it is also much more affordable than the other software although that is secondary.

When the database gets larger and larger, to a point where Microsoft Access cannot handle it, there may be a need to change the database system.

## 2.3 SYSTEM IMPLEMENTATION

### 2.3.1 RP decision support system main page

When users access the RP website, <http://www.eng.nus.edu.sg/LCEL/rpdss>, the RP decision support system main page will appear on



**Fig. 2.6 RP home page**

the users' browsers as seen in Fig. 2.6. Users are able to view the project statement under 'Project Missions', information of the other collaborative universities under 'Our Partners', etc. from the page. However, if their main purpose is to look for RP information, they will have to click on the link 'RP Database' indicated by the arrow in the figure. This will link users to the RP product information main page.

### **2.3.2 RP product information main page**

In the page shown in Fig. 2.7, users can go to other different links. The page will present to the users the following links.

- Browsing of RP data
- Registering as a new RP vendor
- Logging on as RP vendors

The section will start with the explanation of the processes involved in retrieving RP data. Users can click on any of the RP pictures or the subtitles to browse through the information in the database. They can choose to view the following RP information:

- RP Machine
- RP Material
- RP Application

### 2.3.3 Browsing of RP machine data

The web page presented in Fig. 2.8 will be shown when users click on the 'RP Machine' link. As shown in the figure, users can choose to view a specific RP machine such as SLA, SLS, FDM, etc. Users can also choose to view all RP machine types stored in the database. When users choose 'Select all RP machine Type', they will be able to see a list of RP machines as shown in Fig. 2.9.

In the above page, users can select a specific RP machine that is of interest to them. They can also choose to view all RP machines information stored in the database. For instance, when users click on the name 'Sinterstation 2000', essential information about the machine is displayed as seen in Fig. 2.10.

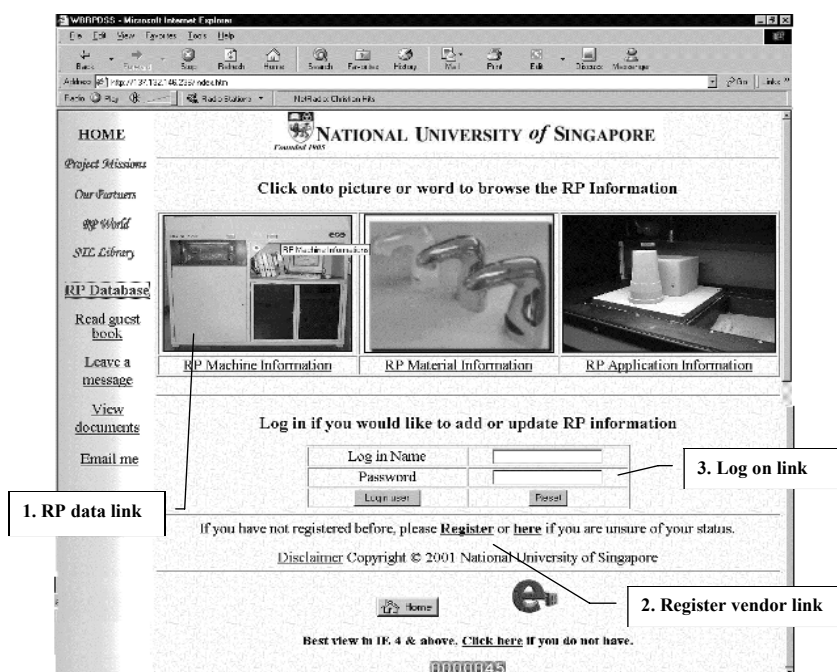


Fig. 2.7 RP equipment page



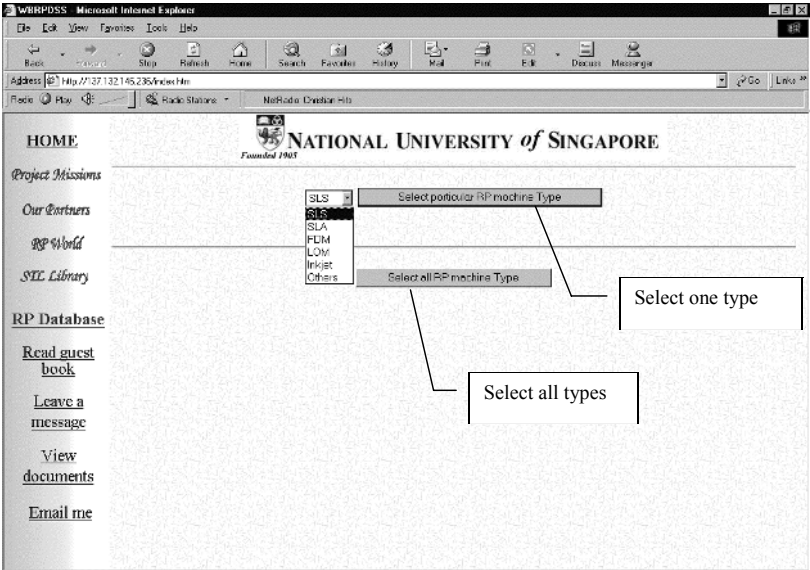


Fig. 2.8 Selecting RP machine type page

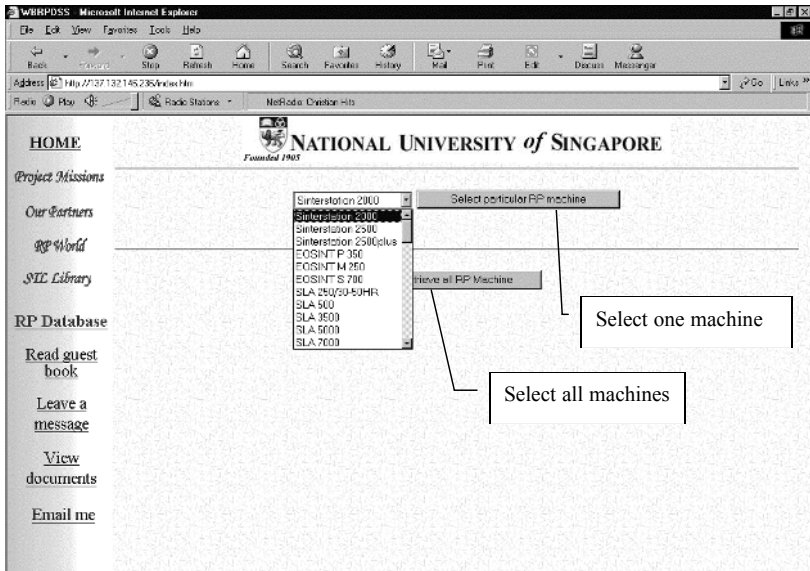
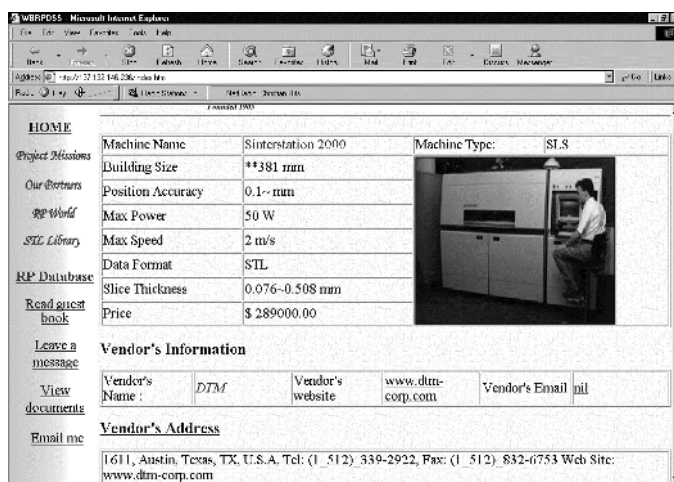


Fig. 2.9 Selecting RP machine page



**Fig. 2.10 Particular RP machine page**

From the figure above, users will have a clear idea of what the machine looks like as there is a picture shown. Other information such as the manufacturer's detail is also given. A direct link to the company's website is provided for users who have enquiries about the machine or wish to purchase the system. The sequence for retrieving RP material and application information are the same as for RP machine.

### 2.3.4 Register as a new RP vendor

In order for users to be registered, they have to click on the link 'register vendor' as shown in Fig. 2.7. Figure 2.11 shows the web page after clicking on the link. However, to be considered as successful applicants, users have to fill up the compulsory fields in the correct format. They are shown in the above figure:

- 'Login Name' – must be more than six but less than ten characters.
- 'Password' – must be more than six but less than ten characters.
- 'Re-type Password' – wording must be the same as typed in 'Password'.
- 'Company Name' – must be less than 255 characters.
- 'Company Email' – must follow the conventional way of logging in an email address, e.g. *abc@smallfoot.com*

After this, the users have a choice of filling up either the 'Company Website' (e.g. *www.smallfoot.com*) or the rest of the columns, information on the company's address if the company does not have a

website; the company can do both if they have a website. Upon successful registration, users will be brought back to Fig. 2.7 with wording notifying users that ‘User has successfully registered as an RP vendor’. After going through the processes of retrieving RP data and registering as a new vendor, the next process that will be introduced is the logging in as an RP vendor.

### **2.3.5 Logging in as RP vendors**

After a successful registration, users can then log in to the RP vendor web page. A sample of the page can be seen in Fig. 2.12.

In the above page, users are given additional features besides browsing through the RP data. Users can add or update RP equipment information into the database. Users can also update their company’s data. The adding of new RP machine information will be discussed in Section 2.3.6.

### **2.3.6 Add new RP machine information**

When users click on the link ‘Add’ in Fig. 2.12, users will be linked to a web page as shown in Fig. 2.13.

In this page, all the textboxes must be filled in correctly. Users have to ensure that the format for each individual column is followed. The formats are:

- ‘Machine Name’ – less than 30 letters.
- ‘Machine Type’ – by default this is the SLS system.
- ‘X Building Size’ – must be numbers and not letters.
- ‘Y Building Size’ – must be numbers and not letters.
- ‘Z Building Size’ – must be numbers and not letters.
- ‘X-Y Axis Accuracy’ – must be numbers and not letters.
- ‘Z Axis Accuracy’ – must be numbers and not letters.
- ‘Name of picture file’ – leave blank if there is no picture available for the machine.
- ‘Max Power’ – must be numbers and not letters.
- ‘Max Speed’ – must be numbers and not letters.
- ‘Data Format’ – must be letters and not numbers.
- ‘Price’ – must be numbers and not letters.
- ‘Min Slice Thickness’ – must be numbers and not letters.
- ‘Max Slice Thickness’ – must be numbers and not letters.

**Registration Form**

Simply fill in the form to register as a Vendor now

Note: Please ensure all columns with \* be filled up

\* Login ID (6 ~ 10) characters

\* Password (6 ~ 10) characters

\* Re-type Password (6 ~ 10) characters

\* Company Name

\* Company

Email

You have a choice of just filling up either company website or company address or both.

Company Website

Address

City

State

Postal Code

Country

Office Telephone

Office Fax

Search Company Information Cancel

Optional: complete one of the two

Compulsory to fill

Fig. 2.11 RP vendor registration page

**NATIONAL UNIVERSITY of SINGAPORE**

Update Company Information

Add Update Add Update Add Update

4. Add and update RP product link

5. Update Company link

Fig. 2.12 RP vendor page

When the users have correctly typed in every textbox, they will be brought back to Fig. 2.12 with words notifying users that ‘You have successfully added the RP machine information.’ If a picture of the RP machine is available, users can click on the link ‘Upload Picture’ as indicated by an arrow in Fig. 2.13. This is to upload a picture of the RP machine to the database (only picture files with the ‘.jpg’ and ‘.gif’ file suffix are allowed).

### **2.3.7 Update RP machine information**

If users click on the link ‘Update’ in Fig. 2.12, they will be led to a web page showing only the RP machines carried by the user’s company as shown in Fig. 2.14. This is to ensure that only one company can edit the RP product information. If users do not have any RP machines belonging to the company stored in the database, there will be nothing shown. The updating sequences are similar to the adding of new RP machine information.

### **2.3.8 Update RP company information**

If users click on the link ‘Update Company Information’ in Fig. 2.12, they will be led to a web page showing only the user’s RP company as shown in Fig. 2.15. The requirement for each textbox is the same as that required in the registering of a new vendor. Users will be linked to Fig. 2.12 with words notifying users that ‘You have successfully updated the RP company information’ upon a successful attempt.

## **2.4 AN EXTENDED DATABASE SYSTEM FOR RP DECISION SUPPORT**

With the information collected on the processes, machines, and materials, the RP database system could now be extended to the web-based RP decision support systems (RPDSS). This evidently is to make the database more friendly and interactive. To make it a DSS there should be some sort of standard or some reference to be compared with to arrive at user-friendly results on the best process associated with the fabrication of the user’s part and useful suggestions on the machine and the materials to be used.

This is where the idea of benchmarking comes into picture. A benchmark serves as a reference for comparison. It basically involves the systematic measuring of a process against a well-established

Microsoft Internet Explorer

Address: http://137.132.146.236/index.htm

National University of Singapore  
Founded 1905

HOME

Project Missions

Our Partners

RP World

STL Library

RP Database

Read quest book

Leave a message

View documents

Email ras

If you have a picture of this RP Machine, Pls click Upload Picture

Note: Please ensure all columns be filled up.

Machine Name:

Machine Type:

Limitation of workpiece size

X Building Size(mm):

Y Building Size(mm):

Z Building Size(mm):

Z Axis Accuracy(mm):

X-Y Axis Accuracy(mm):

Name of picture file (leave blank if no picture):

Machine's Capability

Max Power(W):

Max Speed(m/s):

Data Format:

Price(US\$):

Min Slice Thickness(mm):

Max Slice Thickness(mm):

Submit RP Machine Information

Reset

**Fig. 2.13 Add new RP machine page**

process, and thereby adopting those benchmarked functions or procedures that are more effective. Since benchmarking serves as a reference it should contain all information pertaining to efficiency, accuracy issues, minimum layer thickness, cost, etc. of particular processes, machines, or materials that are selected.

The information on the benchmark database is basically all data that would be collected by fabricating the benchmark parts on various processes, machines, and different materials. These data will clearly depict the efficiency, accuracy, layer thickness, and related information on a particular process, machine, or material. The information would later serve as the references to the customer for the decision support. The purpose of benchmarking and its role in the web-based decision support can be better understood from Fig. 2.16.

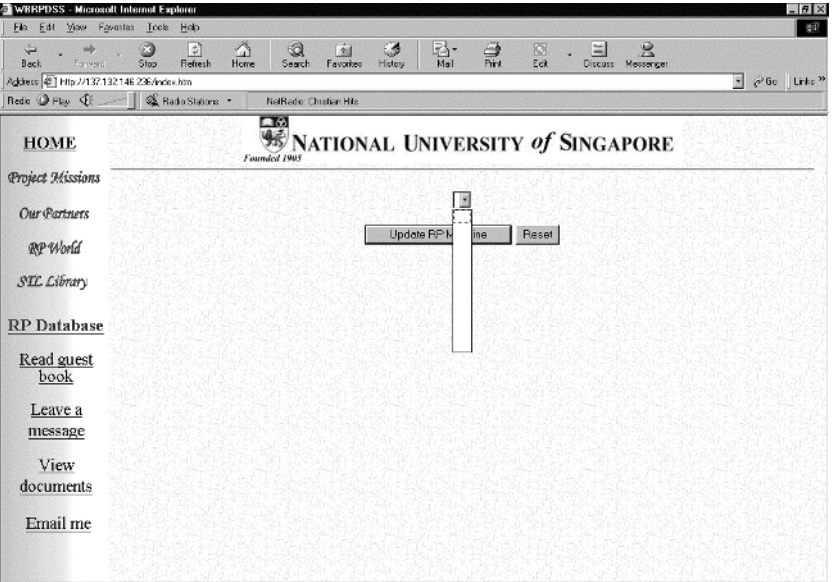


Fig. 2.14 Update RP machine page

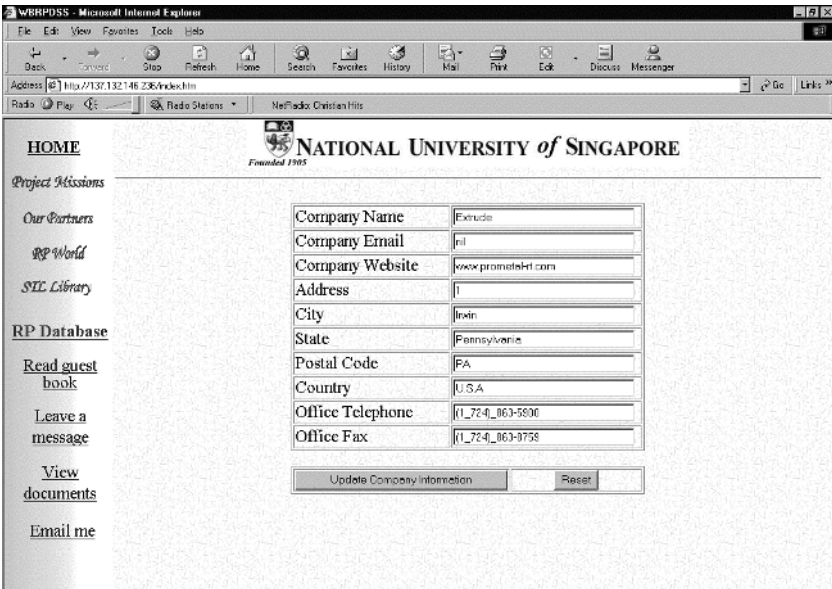


Fig. 2.15 Update RP company page



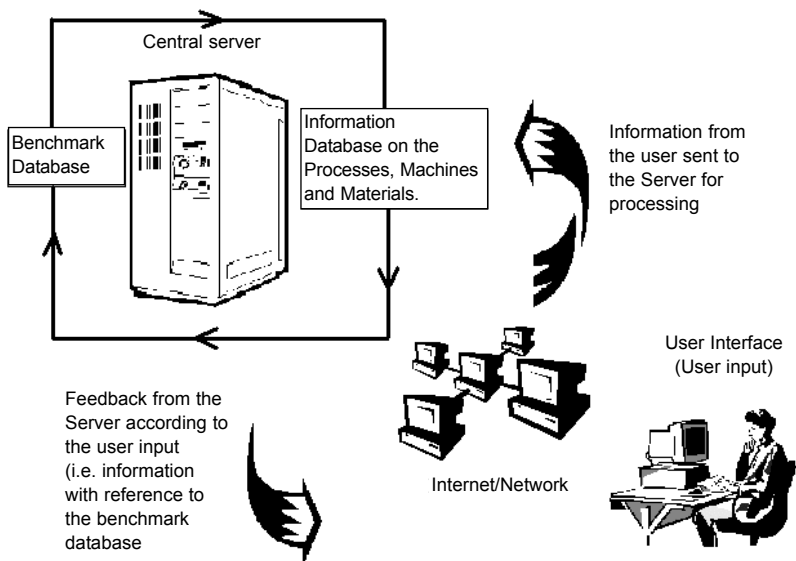
A benchmark database to be set up can either be an integrated one (Fig. 2.16) with the existing database or separated to act as an individual database. The details on benchmark requirements will be described in Chapter 3.

### 2.4.1 The methodology of DSS

The customer/user of the DSS would key in some information according to the requirements from his computer using the user interface (e.g. the customer could key in the information on the part that needs to be fabricated, for which he would want to know the best process, machine, material, and cost). It is appropriate to mention here that the user interface could be customized according to the user's preference.

Once the user submits his information using the user interface, the information is transferred to the central server via the Intranet/Internet for processing. The central server houses both the benchmark database and the process-machine-material database.

At the central server the user information is first checked and verified with the benchmark database for the standard process, machine, and



**Fig. 2.16 Web-based decision support systems**



material that best fit the user's requirement and then that information is transferred to the database to select that particular process, machine, material, and related information in detail. The results are finally fed back to the user for his information. The DSS can also provide the user with contact details of a suitable vendor for the realization of his prototype part. The overall integrated RP database and DSS based on web technology is currently under development.

### **2.4.2 Use of the database**

The use of the database (the machine–material–applications database or the benchmark database) solely depends on the user's requirement, i.e. the user's input will decide the data to be retrieved from a particular database or both. This chapter initially discusses the direct retrieval of information from the machine–material–applications database based on the user's particular choice from the selection.

The use of the benchmark database is particularly vital in its role in the decision support system. In the decision support system the user will not make a direct choice but will input some specifications according to the requirement to arrive at user-friendly results obtained by comparing the benchmark database to the machine–material–application databases. The results obtained here would be more specific as the answers or suggestions to the users' requirements. As mentioned earlier the user interface could be customized to either have a decision support or just retrieve particular information from the database.

There could also be two portals (i.e. two database servers existing in two different locations), which can act together to offer decision support to the customers. The related works on the remote decision support systems could be a topic of further investigation.

### **2.4.3 Scope of future work**

Although the database developed on the rapid prototyping machines, materials, and applications is comprehensive, there is always some updating necessary to keep in track with the present state of the RP industry. Therefore, it is essential to maintain the database, updating all relevant information to meet the growth of the industry. Benchmarks play a dominant role in the DSS and much importance is given in evolving such standard benchmarks. (Chapter 3 deals with such

benchmarks.) Future work such as standardizing the benchmarks is needed to integrate with this database.

## REFERENCES

- 1 Huang, X., Tian, H., and Xu, X. (2000) JSIM database: a web-based database application using XML. In Proceedings of the 38th Annual ACM Southeast Conference, pp. 171–178.
- 2 Diao, X. and Li, C. (1999) Web based database application developed by Intrabuilder. *Mini-Micro Systems*, **20**(5), 363–367.
- 3 Quan, X., Ling, F., and Lu, H. (1999) Supporting web-based database application development. In Proceedings of the 6th International Conference on Advanced Systems for Advanced Applications, pp. 17–24.
- 4 Kulkarni, D. and Marietta, R. B. (1998) Integrated functional and executional modeling of software using web-based databases. *Journal of Database Management*, **9**(4), 12–21.
- 5 Jin, S., Sai, Y., and Du, Y. (1998) Web-based database publishing technology. *Mini-Micro Systems*, **19**(7), 21–25.
- 6 Loeser, H. (1998) Techniques for web-based database applications: requirements, capabilities and architectures. *Informatik Forschung und Entwicklung*, **13**(4), 196–216.
- 7 Lau, T. (1998) Designing, developing, and implementing a web-based, database-driven learning system: the SunTAN experience. In Proceedings of the 1998 IEEE International Professional Communication Conference, IPCC 98, **2**, pp. 85–92.
- 8 <http://www.geocities.com>
- 9 <http://www.astentech.com/scripts/ASP.html>
- 10 <http://www.devguru.com/index.asp?page=/technologies/vbscript/quickref>
- 11 <http://www.aspemporium.com/aspEmporium/codelib/index.html>
- 12 <http://www.cc.utah.edu/~asn8200/rapid.html#LIT>
- 13 <http://www.4guysfromrolla.com>
- 14 <http://www.asptechniques.com>

*This page intentionally left blank*

# Chapter 3

## Rapid prototyping and manufacturing (RP&M) benchmarking

**Y S Wong, J Y H Fuh, H T Loh, and M Mahesh**

Department of Mechanical Engineering, National University of Singapore, Kent Ridge Crescent, Singapore

### ABSTRACT

Rapid prototyping and manufacturing (RP&M) prototypes are increasingly used in the development of new products, spanning conceptual design, functional prototypes, and tooling. Due to the variety of RP&M technologies and processes, resulting in prototypes with quite different properties, planning decisions in the selection of the appropriate RP&M process/material for specific application requirements have become rather involved. Appropriate benchmark parts can be designed for performance evaluation of RP&M systems and processes, and provide helpful decision support data.

Several benchmark studies have been carried out to determine the levels of dimensional accuracy and surface quality achievable with current RP&M processes. Various test parts have been designed for the benchmark study. Most RP&M benchmark studies published to date typically involved fabrication of one sample for each case of material and process. Different companies and machine operators could fabricate the parts. Hence, besides the process and the material, there may be other factors, such as the building style and specific process parameters that may affect the accuracy and finish of the part. It is

noteworthy that comparisons between different processes or between parts built by different companies have generally been based on statistically very small samples.

In RP&M benchmarking, it is necessary to standardize not only the design of the benchmark part, but also the fabrication and measurement/test processes. This chapter presents issues on RP&M benchmarking and attempts to identify factors affecting the definition, fabrication, measurement, and analysis of benchmark parts. The aim is to develop benchmark parts and benchmarking procedures for performance evaluation of RP&M processes/materials in terms of achievable geometric features and specific functional requirements. The RP&M benchmarking design and study will contribute to the development of the planning and decision support software for RP&M processes.

### 3.1 INTRODUCTION

A benchmark is a term originally used by surveyors. It refers to a height that forms a reference or measurement point. Hence, a 'benchmark' is a reference mark for the surveyor (*Webster's New World Dictionary*). The essence of benchmarking is the process of identifying the highest standards of excellence for products, services, and processes, and then making the improvements necessary to reach those standards. It involves systematic measuring of a process against a well-established or performing process, and then adopting and adapting benchmarked functions or procedures that are more effective. The term has also been well used in identifying best practices or processes in manufacturing. Benchmarking has been gaining popularity in recent years. Organizations that faithfully use benchmarking strategies are therefore able to achieve considerable cost and time savings, with quality improvement.

RP&M benchmarking is invaluable for evaluating the strengths and weaknesses of RP&M systems. With the aid of benchmarking, the capability of a specific system can be tested, measured, analysed, and verified through a standardized procedure using standard artefacts. Various RP&M benchmark parts or artefacts have been designed and developed in the last decade, primarily to evaluate the performance of specific RP&M processes. One of the earliest benchmark parts was developed by Juster and Childs (1). This and subsequent benchmark parts have been designed and developed to test geometric accuracy:

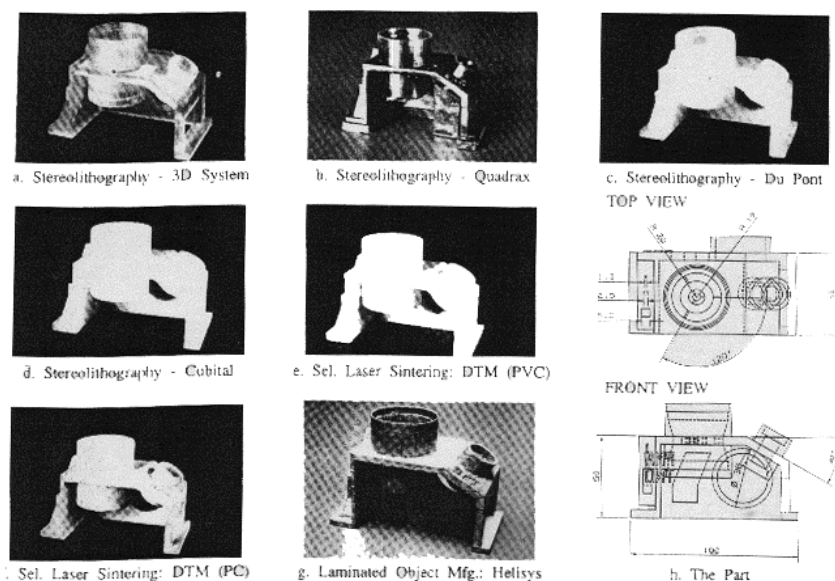
symmetry, parallelism, repeatability, flatness, straightness, roundness, cylindricity, etc.

Today's RP&M application areas extend beyond visualization to functional and final manufactured models. As parts fabricated by different RP&M technologies and processes possess quite different properties, planning decisions to select a suitable RP&M process/material for specific application requirements can be rather involved. Benchmarking can be employed to compare and characterize features across different processes and therefore help to identify suitable RP&M processes for special or new applications. At present, such a generic or common benchmark part is not available for RP&M system builders and users (2).

## 3.2 REVIEW OF RP&M BENCHMARK PARTS

### 3.2.1 Kruth 1991 (3)

An inverted U-frame possessing several geometric features, such as a cylindrical shell, inclined cylinders, pegs, and overhangs, is used as the test part, shown in Fig. 3.1. This benchmark part focuses on the overall performance of the RP&M system. Its largest dimension (100 mm) is relatively small compared with the build size available in most machines.



**Fig. 3.1 Parts produced by different techniques: Kruth 1991 (3)**

### 3.2.2 Gargiulo – 3D Systems 1992 (4)

Targeted to test the in-plane accuracy of SLA machines, the symmetric design of this part in Fig. 3.2 is suitable for the examination of linear accuracy of RP&M parts. Its features are planar and generally it does not test geometric tolerances related to roundness, cylindricity, and concentricity.

### 3.2.3 Lart 1992 (5)

This benchmark part is rich in fine- and medium-sized features as shown in Fig. 3.3. Many of these features, such as the recessed fins and cantilevers, are not easily accessible to a typical coordinate measurement system.

### 3.2.4 Juster and Childs 1994 (1, 6)

The benchmark part has been used to examine both linear accuracy and feature repeatability of RP&M parts built with four main RP&M processes: stereolithography, selective layer sintering, fused deposition modelling, and laminated object manufacturing. The benchmark part shown in Fig. 3.4 incorporates repeated features and dimensions of varying scales for evaluation of the relative merits of the RP

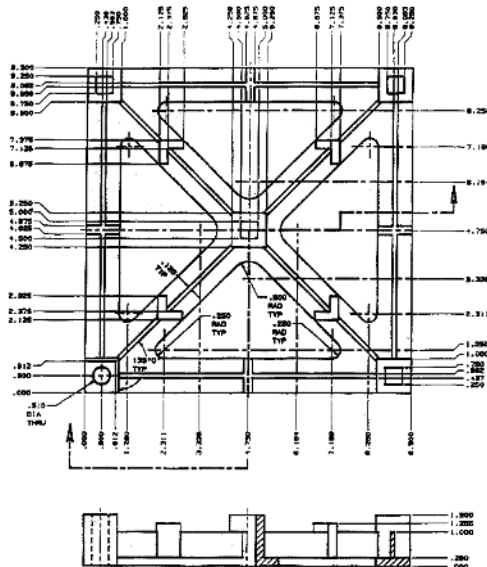
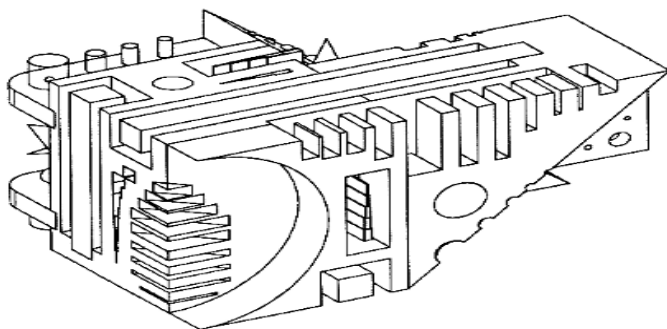


Fig. 3.2 The in-plane benchmark part: Gargiulo 1992 (4)

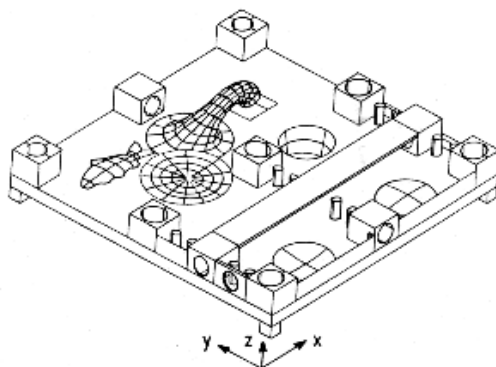
techniques with respect to the accuracy of features in different dimensional scales. Their study shows that the photopolymerization process gives a better performance in the creation of fine features compared with the other processes.

### 3.2.5 Ippolito, Iuliano, and Filippi 1994 (7)

Their first aim was to propose a technique for checking the geometric dimensions and tolerances of RP&M work pieces according to the ANSI-ISO standards. To do this, a well-known user part by 3D System (as shown in Fig. 3.5) was built with different RP&M technologies and with different materials in order to characterize each RP system. The results of their study showed that the user part was unsuitable for assessing the performances of a particular technique in the creation of

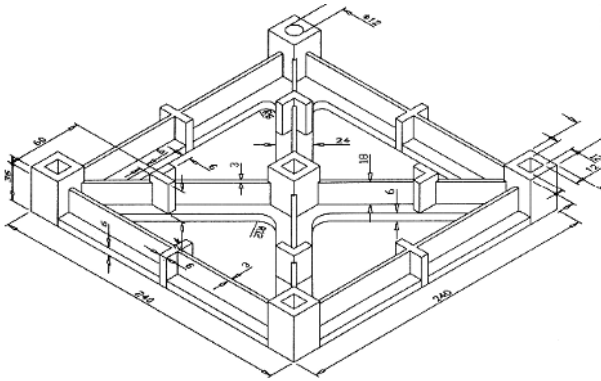


**Fig. 3.3 General view of the model used in comparative study:  
Lart 1922 (5)**



**Fig. 3.4 The benchmark part: Juster and Childs 1994 (1, 6)**





**Fig. 3.5 User part created in metric units: Ippolito, Iuliano, and Filippi 1994 (7)**

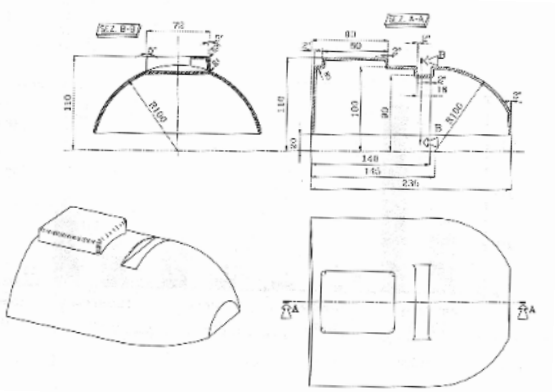
the non-flat surfaces. Therefore a new type of user part shown in Fig. 3.6 was proposed to provide information on the evaluation of accuracy in the reproduction of non-flat surfaces, measurement of roughness, reproduction of very small parts, and creation of relatively thin walls with non-flat edges. The proposed new type of user part consists of a cylinder that merges with a sphere via a gently sloping surface and with appendices extruded towards the inside and outside, bounded by interlinked flat surfaces. However, the measurement of this part is evidently more difficult.

### **3.2.6 Ippolito, Iuliano, and Gatto 1995 (8)**

Using the 3D System benchmark part shown in Fig. 3.5, these authors looked into the development, fabrication, and testing of the benchmark part to investigate dimensional accuracy and surface finish. A new technique of checking the machine quality of an RP&M workpiece according to the ANSI-ISO standards was proposed. The surface of the RP model was also observed by SEM microscope and the study showed that the dimensional accuracy provided by the various RP techniques was generally the same. However, the final result was influenced by the material chosen and the operating parameters.

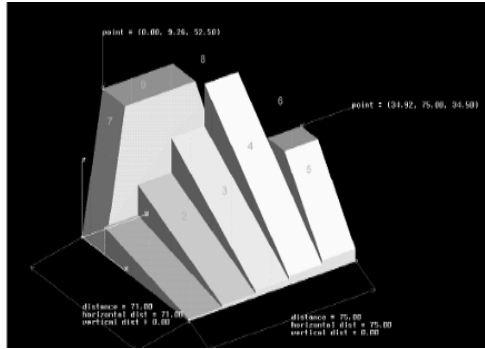
### **3.2.7 Shellabear – EOS Gmbh 1998 (9, 10), and Reeves and Cobb (11)**

A comprehensive study was conducted by Shellabear that involved the fabrication of more than 44 workpieces built using different materials

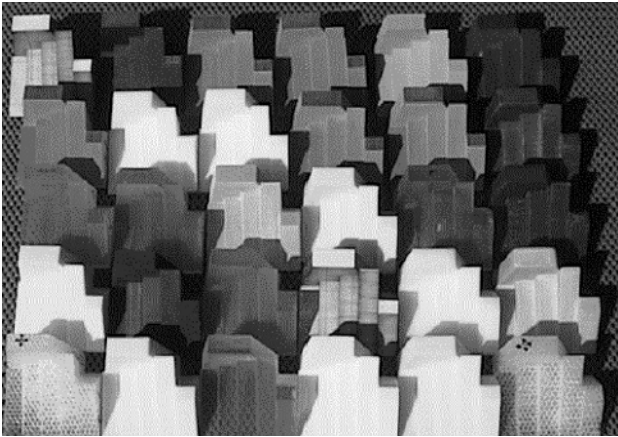


**Fig. 3.6 The proposed three-dimensional user part: Ippolito, Iuliano, and Filippi 1994 (7)**

and RP&M processes based on a proposed benchmark part designed by Reeves and Cobb, as shown in Figs 3.7 and 3.8. Regarding dimensional accuracy, this study could not obtain the accuracy claims (e.g. 0.1 mm to 0.1 per cent) of certain RP&M methods or systems. It appears that in a number of cases, the dimensional accuracy could have been improved by using different scaling factors in the X, Y, and Z-axes of the RP&M system. As the study is based on the same benchmark part used in an earlier study by Reeves and Cobb, the results serve to extend the latter study and provide some indication of progress or changes in the technologies between the two studies. The benchmark part comprises planar surfaces, which include various angles to the building direction (Z) and dimensions in X, Y, and Z. Like most RP&M benchmark studies published to date, typically only one sample of each case (of material and process) was fabricated. The parts were also built by different companies and machine operators. Hence, there may be other factors (besides the process and the material, such as the building style and specific process parameters) that may affect the accuracy and finish of the part. It is noteworthy that the author cautions that comparisons between different processes or between parts built by different companies are therefore based on statistically very small samples and should be treated with caution. In addition, although the same benchmark part was used, the measurement results could not be directly correlated with those of Reeves and Cobb due to different measurement methods.



**Fig. 3.7 Geometric benchmark part by Reeves and Cobb (11)**



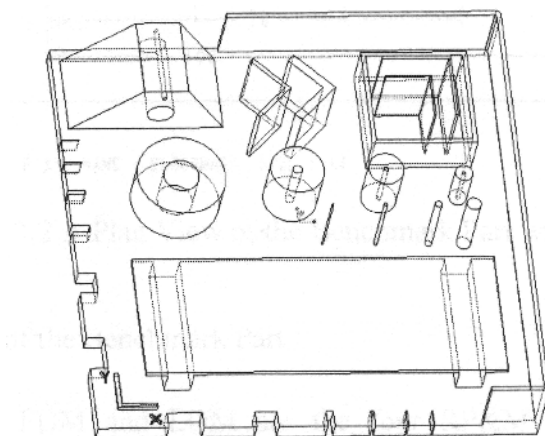
**Fig. 3.8 Test parts from different RP&M processes: Shellabear (9, 10)**

### **3.2.8 Fen and Dongping 1999 (12, 13)**

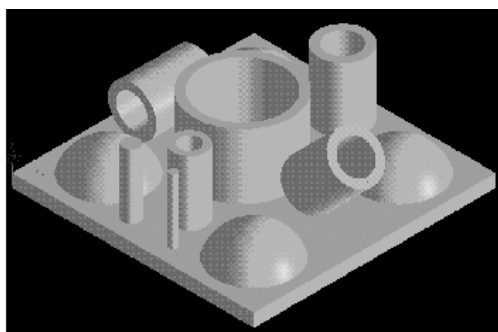
The research efforts of Xu Fen and Shi Dongping on benchmarking aim to develop practical and more generic benchmark parts that can provide data for decision support, and enable comparative performance analyses/evaluations of different RP&M processes/systems (see Figs 3.9 and 3.10).

## **3.3 TYPES OF RP&M BENCHMARKS**

Benchmarks for RP&M processes and systems can be classified into three main types: geometric, mechanical, and process.



**Fig. 3.9 Benchmark part: Fen (12)**



**Fig. 3.10 Benchmark for geometric accuracy: Dongping (13)**

### 3.3.1 Geometric benchmark

A geometric benchmark is used to check the geometric and dimensional accuracy of the prototype. The desired accuracy requirement is often defined in terms of established standards, for example, the ANSI-ISO or German standard DIN 16901 for moulded parts (9). Several geometric benchmark parts have been reported. Typical geometric features incorporated in these geometric benchmark parts are circular holes, cylinders, thin walls, slots, and squares. Better-known benchmark part designs are listed in Table 3.1, which indicates if certain geometric features to be checked or measured are adopted by these benchmark parts for performance evaluation of the RP&M process where the part has been built.

### 3.3.2 Mechanical benchmark part

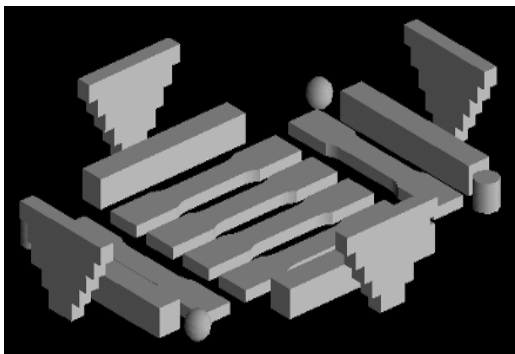
A mechanical benchmark part as shown in Fig. 3.11 is designed to provide components that can be used to characterize the mechanical properties of the RP&M part. The components can be fabricated simultaneously and later separated to test individual mechanical properties.

#### 3.3.2.1 Mechanical properties

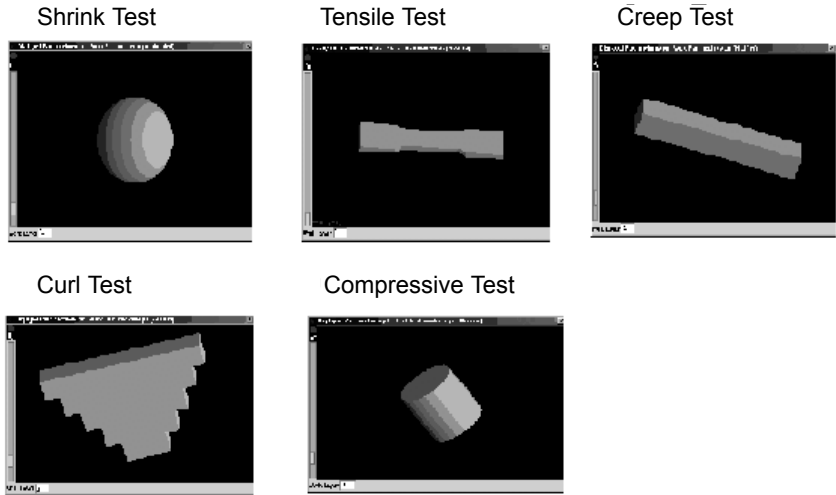
The main mechanical properties to be evaluated include shrinkage, tensile and compressive strengths, curling, and creep characteristics. The obvious influence of shrinkage and curling are geometric distortions affecting dimensional inaccuracy. The mechanical properties cited here are based on its importance; however, more properties can be included and tested according to the specific requirements of the prototype. As can be seen in Fig. 3.12, various components are to be removed from the benchmark part for testing to determine the mechanical properties.

We can generally use the standards that are used by the ASTM for testing a particular property. These are listed in the following section. (Note that an ASTM subcommittee E28.16 was earlier formed to look into the development of procedures for testing the mechanical properties of RP-fabricated parts. However, the subcommittee no longer exists.) RAPTIA, which is a European Thematic Network of research organizations, universities, and industries working with rapid tooling, has developed the RAPTIA benchmark for tensile testing (14).

The typical mechanical properties to be tested include the following.



**Fig. 3.11 Mechanical benchmark part**



**Fig. 3.12 Components from mechanical benchmark part**

- Tensile strength – the ability of the material to withstand tensile loading. ASTM specifies standards depending on the material used for the test. For example:
  - Tensile Testing of Metals – ASTM E 8
  - Tensile Testing of Plastics – ASTM D 638
  - Tensile Testing of Polymer Matrix Composites – ASTM D 3039
  - Tensile Testing of Metal Matrix Composites – ASTM D 3552
- Compressive strength – the ability of a material to withstand compressive loading. ASTM also specifies the standard testing method for the determination of the compressive property.
  - Compression Testing of Metals – ASTM E 9
  - Compression Testing of Rigid Plastic – ASTM D 695
  - Compression Testing of Rigid Cellular Plastics – ASTM D 1621
- Shrinkage – amount of contraction or dimensional change due to the shrinking of the material. ASTM mentions a few testing methods but they can only be treated as a reference for RP purposes (for example ASTM C531 Standard Test Method for Linear Shrinkage and Coefficient of Thermal Expansion of Chemical-Resistant Mortars, Grouts, Monolithic Surfacing, and Polymer Concretes).
- Curl – tendency to twist or coil – the degree of curvature of a material sheet when exposed to varying conditions of humidity. ASTM specifies standards for the measurement of curl in

carbon/carbonless paper, tapes, copper, and copper strips; hence, the appropriate standard can serve as a guide when used to measure the curl of an RP according to its material.

- Creep – time-dependent permanent deformation that occurs under stress, which for most materials, is important only at elevated temperatures. ASTM E139 specifies the method to determine the amount of deformation as a function of time (creep test) and the measurement of the time for fracture to occur when sufficient load is present (rupture test) for materials under constant tension loads at constant temperature. It also describes the essential requirements for the test equipment. ASTM also specifies standard testing methods according to the material used.
  - Creep/Rupture of Plastics ASTM D 2990
  - Creep/Stress Rupture of Metallic Materials ASTM E 139

### **3.3.3 Key attributes for benchmark parts**

When considering a benchmark part, it is also necessary to take into account the application of the prototype, which can fall into one of the following categories (15).

- Design prototype: It serves first and foremost for visual evaluation, such as the aesthetic and ergonomic aspects, while the geometric accuracy or mechanical requirements are normally not important considerations.
- Geometric prototype: This is employed for testing accuracy, form, and fit of the part.
- Functional prototype: It represents a set of features, which allow evaluation of some functional aspects. A functional prototype is usually a subsystem of a product.
- Technical prototype: This covers all functional aspects of the part and can be used as such, although the final manufacturing process may be different from the RP&M process. The technical prototype may also consist of different materials.

The design of the benchmark part should cover the above aspects of the prototypes, i.e. whether it is a design, geometric, functional, or technical prototype.

## **3.4 TOWARDS GENERALIZED BENCHMARK PARTS**

The primary performance indices for evaluating RP&M systems are speed, cost, and dimensional accuracy. Current approaches for

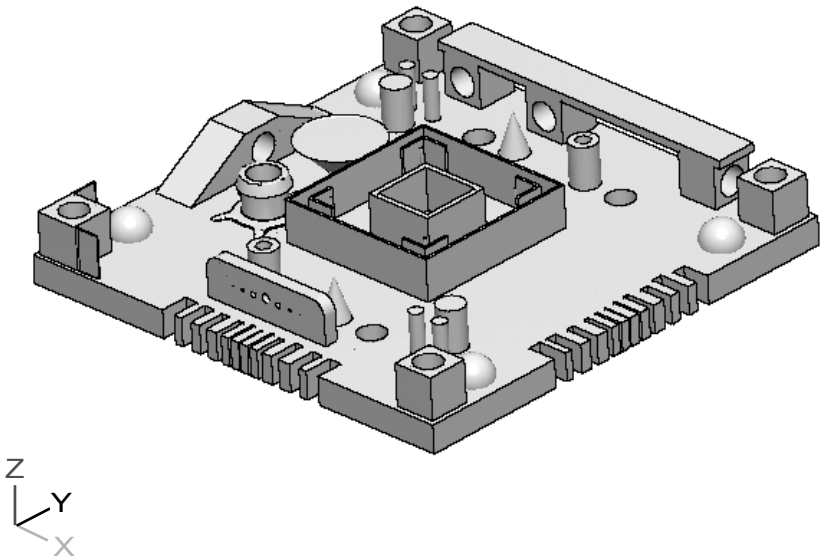
analysing these factors have consisted of defining and building benchmark parts. Several user organizations have developed various benchmark parts for evaluating specific RP&M systems or particular aspects of an RP&M process. These benchmark parts tend to be system or process dependent and may not provide meaningful or comparative data across applications, systems, or processes. In addition, the lack of standardized procedures for building and measuring the benchmark parts can incur significant variations in the outcomes of the performance evaluation conducted on the benchmark parts, as has been reported by Shellabear (9, 10). There are up to nearly 20 proposed and used benchmark parts available to the RP&M industry (2). This indicates the need to search for suitable benchmark parts that can be used across the various RP&M systems/processes. It also implies that such a generalized benchmark part will not be straightforward to design and develop.

This section discusses and presents proposals towards generalized benchmark parts and associated standardized procedures for the fabrication, testing/measurement, and evaluation of the parts. The RP&M system performance evaluation will be assumed to be based upon a benchmark part consisting of three-dimensional part features built in a variety of sizes, locations, and orientations. The mechanical properties will be based upon a mechanical benchmark part and the process of fabrication will be based on the process benchmark. Test, measurement, and analytical procedures for the various feature characteristics are defined.

### **3.4.1 Proposed geometric benchmark**

The proposed geometric benchmark part is shown in Fig. 3.13, which aims to incorporate key shapes and features that are currently employed in better-known benchmark parts. Figures 3.14 and 3.15 show the top and front views. The benchmark includes geometric features, such as freeform surfaces, that are increasingly required or expected of RP&M processes/systems. The applicability of the proposed design as a generalized benchmark is discussed with respect to its suitability for evaluation of the four widely used RP&M processes: stereolithography (SLA), selective laser sintering (SLS), fused deposition modelling (FDM), and laminated object modelling (LOM).





**Fig. 3.13 Proposed geometric benchmark part**

#### ***3.4.1.1 Functions of the proposed geometric features***

Table 3.1 summarizes the functions of the main geometric features in the proposed benchmark part design. A comparison of key features with better-known benchmark parts is given in Table 3.2. The geometric features have been designed with a specific purpose. These are discussed below.

The square base, which supports all the standing features, is itself a test for straightness and flatness. The dimension of the square base was chosen to be  $170 \times 170 \times 5$  mm to account for the average build size of most machines. The eight cubes are used to test for linear accuracy, straightness, flatness, parallelism, and repeatability. The relative distance and parallelism can be measured between the faces that are symmetrical. The cylindrical holes in the cubes are employed to test for accuracy, roundness, cylindricity, and repeatability of radius. Two cylindrical holes have axes in the  $X$ -direction, two have axes in the  $Y$ -direction, and others have axes in the  $Z$ -direction. The holes in different axes are for accuracy and concentricity tests in the particular axis. The  $X$ -axis and  $Y$ -axis cylindrical holes aim to test for overhang effects.

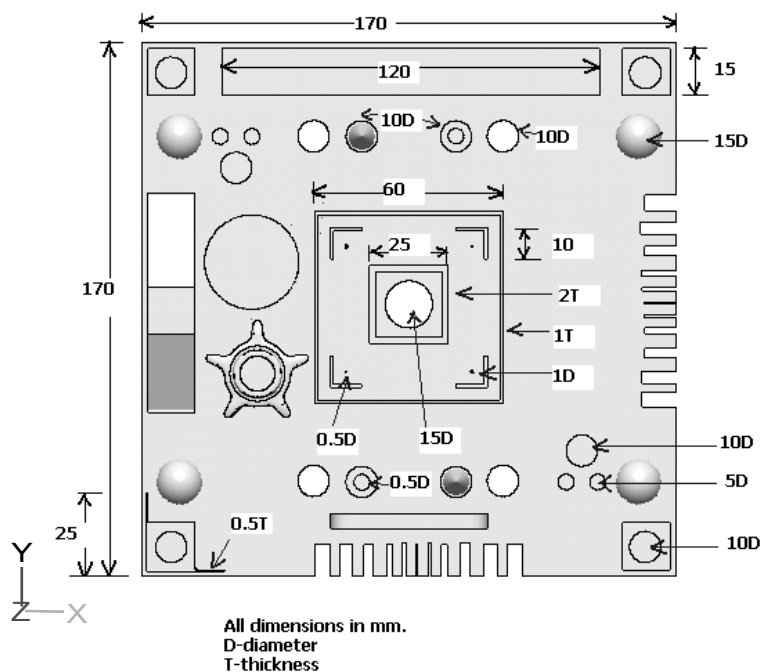


Fig. 3.14 Geometric benchmark – top view

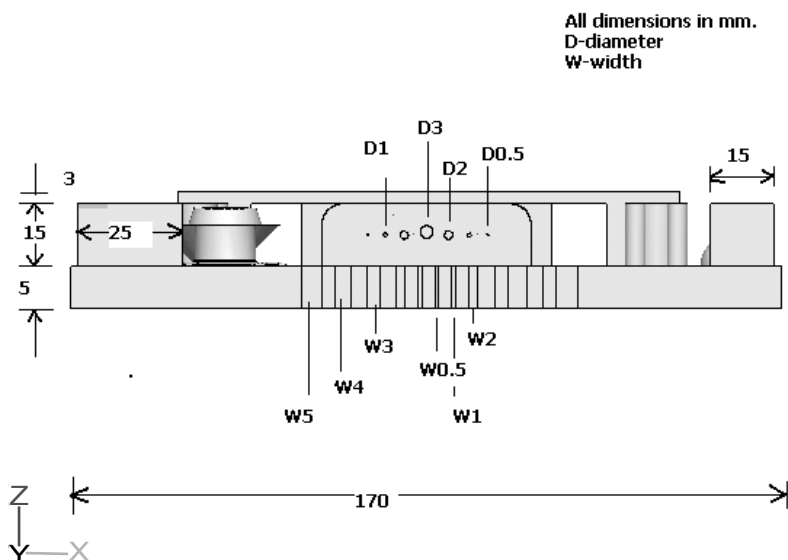


Fig. 3.15 Geometric benchmark – front view

The flat beam  $120 \times 15 \times 3$  mm is a test for the machine to build crossbeams. The beam is divided into two unequal spans by a square boss to determine the effectiveness of the machine to build short- and long-span beams supported at the ends. The four spheres that are placed symmetrically test for a continuously variable sloping profile of the surface, with respect to relative accuracy and repeatability. The spheres are also a test for symmetry.

The two hollow squares towards the centre test for straightness, flatness, and smoothness of the surface. They are also a test for building smooth walls and are suitable for determining the surface finish. The solid cylinders test for roundness, cylindricity, and repeatability. Their relative distance could also be used for positioning accuracy test. The hollow cylinders more or less serve the same purpose, but in addition they are a test for accuracy, cylindricity, and concentricity. The pair of cones is a test for a sloping profile at a specific taper angle.

The slots on the square base are used to test the ability of the machine to build fine slots. The slots vary in dimension so as to test the accuracy of building slots of varying dimensions. A set of slots is placed on one side and another set on an adjacent perpendicular side. The reason for such a design is to test the build efficiency of the slots in two mutually perpendicular directions. The five circular holes on the square base test for symmetry and also act as a measure of the relative distance. Circularity and concentricity can also be studied. The wedges that are placed on either side of the cube are used to measure an inclined plane.

On the proposed benchmark part, there are a number of features of critical importance and these are termed pass/fail features. The pass/fail features include the thin walls of 0.5 mm thickness, vertical solid cylinders of 0.5 mm diameter, circular holes of 0.5 mm diameter, slots of width 0.5 mm, and freeform features. The pass/fail features test the ability of the RP&M system/process to build such features. With only a visual comparison it is possible to provide some measure of the efficiency of the RP&M system to build such features, even without the actual measurements being taken.

**Table 3.1 A summary of the proposed geometric features and their purpose**

Features	Purpose	Number and size
Square base	Flatness and straightness (a base for the other features)	1 (170 x 170 x 5 mm) (Size likely to fit build size of most machines)
Cube	Flatness, straightness, linear accuracy, parallelism, and repeatability	8 (15 x 15 x 15 mm)
Flat beam	Overhang, straightness, and flatness	1 (120 x 15 x 3 mm)
Cylindrical holes (Z-direction)	Accuracy, roundness, cylindricity, and repeatability of radius	4 (10 mm diameter)
Cylindrical holes	Accuracy, roundness, and concentricity	X-direction: 2 (10 mm diameter) Y-direction: 2 (10 mm diameter)
Spheres	Relative accuracy, symmetry, and repeatability of a continuously changing sloping surface	4 (15 mm diameter)
Solid cylinders	Roundness, cylindricity, and repeatability	4 (5 mm diameter) 2 (10 mm diameter)
Hollow cylinders	Accuracy, roundness, cylindricity, and repeatability of radius	2 (outer diameter 10 mm; inner diameter 5 mm)
Cones	Sloping profile and taper	2 (base diameter 10 mm)
Slots	Accuracy of slots, straightness, and flatness	11 (length 10 mm, height 5 mm, varying width) arranged in two rows
Hollow squares	Straightness and linear accuracy, also thin wall built	2 (first: 25 x 25 x 15 mm, wall thickness: 2 mm; second: 60 x 60 x 15 mm, wall thickness: 1 mm)
Brackets	Linear accuracy, straightness, and angle built	4 (length 10mm, height 15mm, width 10mm, thickness 1mm)
Circular holes	Cylindricity, relative position, roundness, and repeatability	5 (one 15 mm diameter; four 10 mm diameter)
Mechanical features	Efficiency of machine to build special features	Fillet, chamfer, blending, freeform features
Pass/fail features	Ability of machine to build certain features	Thin walls, thin slots, slim cylinders, small holes

**Table 3.2 Comparison of selected geometric benchmark part designs**

Geometric features	Proposed benchmark	Feature evaluation	Juster and Childs' benchmark (6)	3D System's benchmark	Dongping's benchmark (13)	DTM's benchmark (geometric accuracy)	Fen's benchmark (12)
Solid cylinders	✓	Circularity and cylindrical profile	—	—	✓	—	✓
Hollow cylinders	✓	Cylindrical and hollow profile	✓	✓	✓	—	✓
Hemispheres	✓	Roundness and slope	✓	—	✓	—	—
Cones	✓	Slope, sharpness, and staircase effect	—	—	—	—	—
Thin walls	✓	Efficiency test (flatness and smoothness)	✓	✓	✓	✓	✓
Brackets	✓	For measurement of angle build	—	✓	✓	—	✓
Square	✓	Surface profile, smoothness and sharp edges)	✓	✓	✓	✓	✓
Vertical holes	✓	Profile of vertical sharp holes	✓	✓	✓	—	✓
Horizontal holes	✓	Ability to build overhangs	✓	—	✓	—	—
Chamfered ends	✓	Slope profile and angle	—	—	—	—	—
Slots	✓	Accuracy in building slots	✓	✓	✓	✓	✓
Overhang	✓	Ability to build support structures	✓	—	✓	—	✓
Fillet	✓	Ability to build such features	✓	✓	—	—	—
Pass/fail features	✓	Ability to build such features	✓	—	—	—	✓
Freeform features	✓	Test building of freeform features	✓	—	—	—	—

✓ denotes that the particular feature has been included; — denotes that the particular feature has not been included.

### 3.5 PROCESS BENCHMARKS

Besides identifying an appropriate benchmark part design, care has also to be taken with regard to the fabrication and the process related

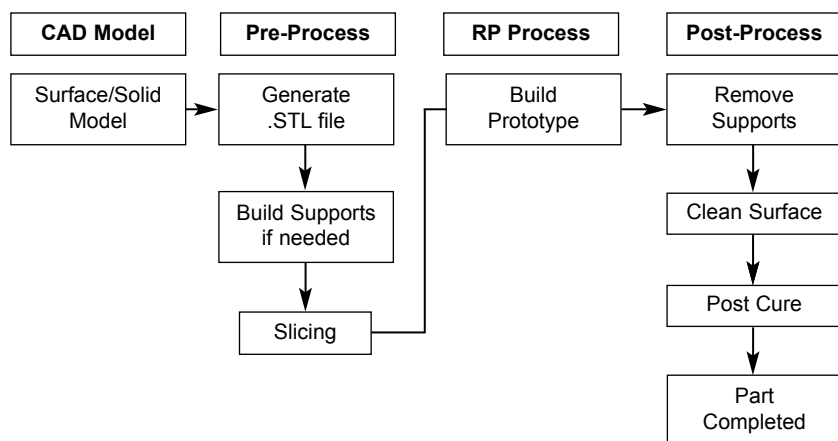
to it. Hence, process benchmark is another important consideration. The process planning steps include: part orientation, support structure design, slicing of the part into two-dimensional contours, path planning, and process parameter selection. Research work on the process planning has been carried out in the past by many researchers, of which that by Rosen (16) is notable.

The goal of the process is to improve the accuracy and surface finish, and obviously reduce the build time. The part orientation and layer thickness for the part slicing are factors that need attention in evolving with a process benchmark.

There are also several other parameters to be controlled during the fabrication of a part, such as hatching, scan-speed, laser, etc. Obviously the best method for the fabrication of the RP&M part and the measurement technique for it also need to be identified.

### 3.5.1 RP&M process chains

The RP&M process generally includes the entire fabrication stage spanning from CAD modelling to final fabrication of the part as shown in Fig. 3.16. To benchmark the entire process, each stage of the process has to be meticulously examined so as to select the best parameters and identify the best practice (17). Key considerations are listed in Appendix A.



**Fig. 3.16 RP&M process chain**

### **3.5.2 Measurement of RP&M parts (2, 18)**

Important considerations in the evaluation of RP&M system performance are consistency and standardized approaches and techniques. Current techniques for the measurement of parts produced by RP&M processes are not too different from those for machined parts. Specific issues are primarily due to the layered nature of the RP&M parts, but other factors such as surface roughness must also be considered (2). Similar issues occur in other measurement situations when parts have irregular surfaces, such as for parts made by casting or powder metallurgy. A standardized methodology/procedure helps to ensure consistency and reduce variability.

A co-ordinate measuring machine (CMM) is well suited for the measurement of the RP&M parts because of its versatility and speed (18). Most CMMs have high accuracy compared with other measurement methods and can be programmed to carry out a variety of automatic measurements, ranging from simple to complex. A CMM determines the measured dimensions and shape errors, namely flatness, parallelism, angularity, straightness, and roundness. Some basic measuring instruments such as the vernier calliper and screw gauge with high accuracy are also used in the measurement of RP&M parts. However, these have limitations in the part size that can be measured.

Using standardized measurement techniques (such as with the CMM in the measurement of RP&M parts) would facilitate comparison of the geometry of RP&M parts in general, and incorporating such standardized measurement techniques in the RP&M industry would help to achieve a consistent and wider comparison of the RP&M parts across the industry with reference to the geometry.

### **3.5.3 Considerations for process benchmarking (19)**

Process planning consists of generating the information required to fully specify the manufacturing steps needed to build a particular prototype. In general, process planning for rapid prototyping processes consists of three distinct sequential steps (19):

- Geometric decomposition
- Operation planning
- Machine code generation

In geometric decomposition, the part model is divided into manufacturable sections. Parameters, such as tool paths, required to build each of these sections of the part are generated during the operation planning. Finally, these parameters are translated into machine commands, which run the part-building machine. Each of these steps is usually highly process dependent.

### ***3.5.3.1 Planning for rapid prototyping processes***

RP&M process planning usually starts with a geometric model in the STL file format. The STL format represents the part surface as a set of triangular facets, each of which also has a normal vector to indicate the side that is outside. Because the surface is faceted, some geometric information is lost. Curved surfaces in particular are no longer easily identified. Large numbers of facets are needed to represent curved surfaces accurately and this typically results in large file sizes. There can also be model quality issues if there are gaps where facets do not join exactly.

Most RP&M processes build parts by stacking planar layers of uniform thickness. There is a particular sequence of planning and operation for the process in general. Given the part model, the steps required are as follows.

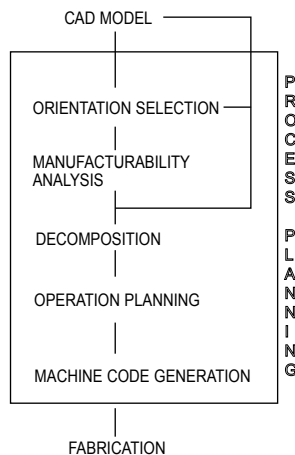
- The first step is to determine where support structures are required, if necessary.
- The model is decomposed into layers.
- Layers are usually planar and of a uniform thickness. Curved-layer LOM is an exception where uniform-thickness curved layers are used (19).
- The two-dimensional cross-sections represent the areas that must be filled with part and support material to form each layer. A path-planning algorithm then generates a suitable scanning pattern that will fill these areas.
- The final step is to build the part by executing the deposition patterns, layer-by-layer, until the entire part is fabricated.
- The support material is removed to recover the finished part.

The planning process is illustrated in Fig. 3.17; the input is a CAD model of the part to be fabricated. Given the model, the first step is to determine the build orientation for the part. There will typically be many potential build orientations, and these must be analysed for



manufacturability so that the best orientation can be identified. This will typically be an iterative process where each orientation is evaluated in turn. If there are no manufacturable orientations, then it may be necessary to modify the original CAD model to make it manufacturable. Alternatively the part can be built to approximate the model as closely as possible, given the limitations identified in the manufacturability analysis. Once the build orientation has been selected, the model must be decomposed into manufacturable sections after which a set of operations must be generated to manufacture each section. This involves deposition and path generation as well as specification of curing, cooling, and preheating operation parameters. An operation sequence will be generated describing operations and the order in which they should be performed in order to build the part.

Typical parameters involved in the part fabrication include process control parameters, materials to be deposited, curing, cooling time, and preheat power levels. The process planning focuses primarily on the decomposition, operation planning, and machine code generation steps. Machine code generation, which can be very tedious, is relatively easy to automate since it consists mostly of translating operation specifications, particularly deposition and path generation, into machine-readable formats. Decomposition and operation planning are much more difficult tasks and are targets for automation because they are time consuming and require detailed process knowledge and experience.



**Fig. 3.17 Process planning**

### 3.5.3.2 *Manufacturability analysis*

The goal of manufacturability analysis is to determine if a given geometric model can be built in a particular orientation with minimal geometric approximations or defects due to limitations of the manufacturing process. Because of the need to determine if a part can be built near to the required shape, manufacturability analysis varies among typical RP&M processes. Except in special cases, commercial RP&M processes cannot build the exact part geometries and therefore there will always be some degree of geometric approximation. The primary example of this is in the fabrication of inclined surfaces, where RP&M processes produce a stair-step effect rather than a smooth surface.

Because of this approximation, the manufacturability analysis for RP&M processes typically concerns the search for a build orientation where the approximation error is minimized. A variety of methods can be used for this optimization. The volumetric error can be estimated by computing the geometry of the part that will be built and comparing it with the original model. Surface smoothness can be calculated based on the angle of a surface relative to the build direction. Surfaces that are parallel or perpendicular to the build direction are usually best.

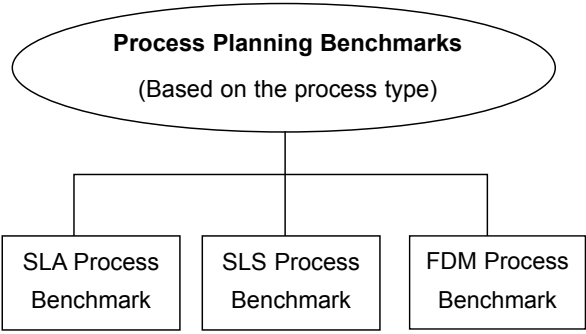
The benchmarking of a particular RP&M process thus involves the aforementioned issues and arriving at standards that correspond to the process that would finally be used to fabricate the model.

## 3.6 FRAMEWORK FOR PROCESS BENCHMARKING

As discussed in the preceding sections, process planning in RP&M generally encompasses the following considerations.

- |                           |   |                          |
|---------------------------|---|--------------------------|
| • Geometric decomposition | } | ➡ <b>Pre-processing</b>  |
| • Operation planning      |   |                          |
| • Machine code generation |   |                          |
| • Fabrication             |   | ➡ <b>Fabrication</b>     |
| • Removal of Supports     | } | ➡ <b>Post-processing</b> |
| • Measurement             |   |                          |
| • Inspection              |   |                          |

The process benchmark is process dependent and has to be defined with respect to each specific process, as illustrated in Fig. 3.18. It should



**Fig. 3.18 Process benchmarks**

encompass the best practice/processing procedure for the specific RP&M process according to its application by the RP&M industry.

**3.7 RP&M MATERIAL SELECTION (20)**

Material selection is an important factor when considering the fabrication of a part by RP&M. It also may not necessarily be an independent consideration in RP&M process planning. Materials used in rapid prototyping are limited and dependent on the process, but the range and properties available are growing quickly (20). Although there may be many indirect methods involving a variety of materials, the rapid prototyping processes can work directly only on a limited range of materials. Table B1 in Appendix B shows such principal direct building materials. Some of the materials used in rapid prototyping are also listed in Table B2.

**3.8 INTEGRATION WITH DATABASE FOR DECISION SUPPORT**

Ultimately, the data obtained from the standardized measurement and testing procedures of benchmark parts fabricated from a benchmarked RP&M process will have to be processed for the evaluation of a particular process and the system based on that process. A database that is easily accessible by Internet will be very useful as a central repository of such data. Such a database can be used to capture important information of RP&M processes and systems for use by decision support tools or systems to select suitable systems or processes to meet specific objectives, as illustrated in Fig. 3.19.

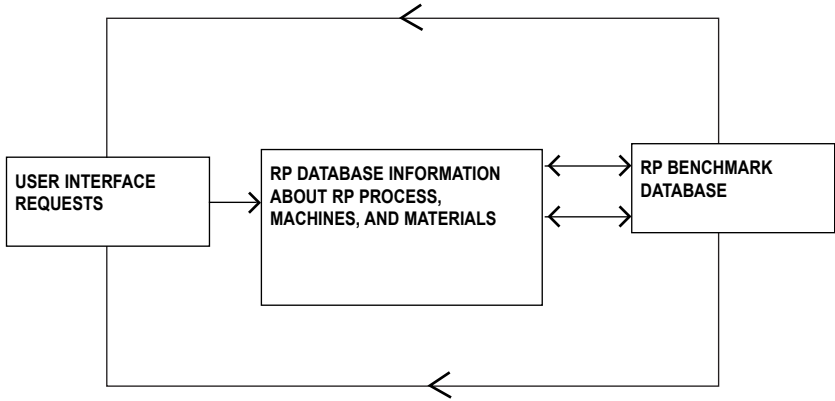


Fig. 3.19 Flow diagram

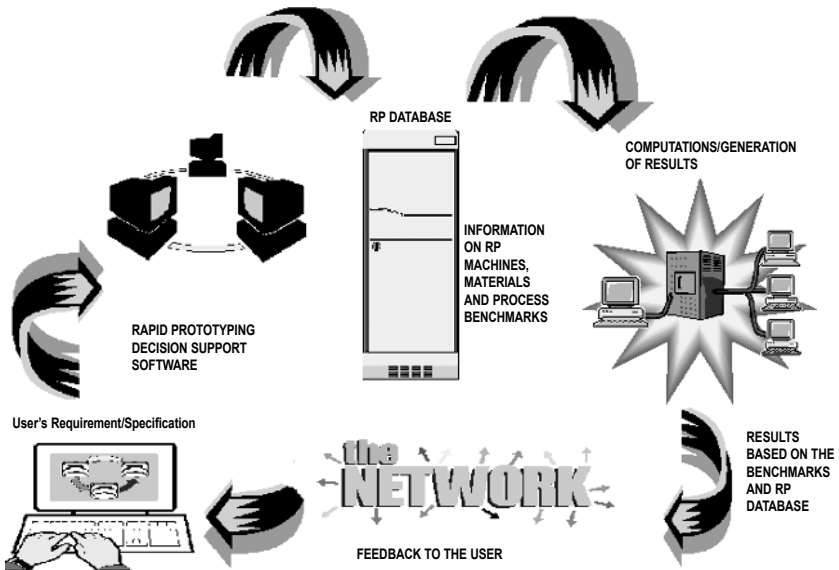


Fig. 3.20 Flow diagram for the RP&amp;M DSS software

Figure 3.20 depicts such a web-based database integrated to a decision support system (DSS) (See Chapter 2). The database serves as a central repository for information pertaining to rapid prototyping and manufacturing. It includes detailed information regarding machines, materials, applications, vendor information, etc.

The rapid prototyping and manufacturing decision support system (RP&M DSS) helps to find the best method to manufacture a physical

or prototype model. The RP&M DSS can provide information on RP&M machines, RP&M materials and follow-up processes. It also aims to capture data from benchmarked parts to provide decision support information that can assist the user in selecting a suitable process to meet specific requirements. The RP&M DSS is web-based to enable the RP&M users worldwide to have a common gateway to all RP&M-related endeavours.

### 3.9 CONCLUSION

RP&M benchmarking would greatly facilitate decision making to identify and determine a suitable RP&M process or system for the fabrication of an RP&M part that meets specific requirements. Various standardization issues have been given, presented, and discussed with regard to appropriate geometric benchmarks, mechanical benchmarks, and the process benchmarks. The geometric and mechanical benchmarks could be suitably fabricated by different RP&M processes and inspected/tested through best practices or controlled processes defined by the process benchmarks and the results captured and stored in a web-based database that serves as an easily accessible central database. This benchmark database will then be used by appropriate RP&M decision support systems to identify suitable RP&M machines, materials, and processes to meet specific requirements. Additional information on suitable vendors could also be generated.

### REFERENCES

- 1 Juster, N. P. and Childs, T. H. C. (1994) Linear and geometric accuracies from layer manufacturing. *Annals of the CIRP*, **43**(1), 163–166.
- 2 Jurens, K. K. (1999) Standards for Rapid Prototyping Industry, *Rapid Prototyping Journal*, **5**(4), 169–178.
- 3 Kruth, J. P. (1991) Material increment manufacturing by rapid prototyping techniques. In 41st General Assembly of CIRP, Stanford, California, *CIRP Annals*, **40**(2), 603–614.
- 4 Gargiulo, E. P. (1992) Stereolithography process accuracy: user experience. In Proceedings of 1st European Conference on *Rapid Prototyping*, University of Nottingham, UK, 6–7 July, pp. 187–207.
- 5 Lart, G. (1992) Comparison of rapid prototyping systems. In Proceedings of 1st European Conference on *Rapid Prototyping*, University of Nottingham, UK, 6–7 July, pp. 243–254.

- 6 Juster, N. P. and Childs, T. H. C. (1994) A comparison of rapid prototyping processes. In Proceedings of 3rd European Conference on *Rapid Prototyping and Manufacturing*, University of Nottingham, UK, 6–7 July, pp. 35–52
- 7 Ippolito, R., Iuliano, L., and Filippi, A. (1994) A new user part for performance evaluation of rapid prototyping systems. In Proceedings of 3rd European Conference on *Rapid Prototyping and Manufacturing*, University of Nottingham, UK, 6–7 July, pp. 327–339.
- 8 Ippolito, R., Iuliano, L., and Gatto, A. (1995) A benchmarking of rapid prototyping techniques in terms of dimensional accuracy and surface finish. *CIRP Annals*, **44**(1), 157–160.
- 9 Shellabear, M., EOS GmbH (1998) Model manufacturing processes – state of the art in rapid prototyping. *RAPTEC*, Task 4.2, Report 1.
- 10 Shellabear, M., EOS GmbH (1999) Benchmark study of accuracy and surface quality in RP&M models. *RAPTEC*, Task 4.2, Report 2.
- 11 Reeves, P. E. and Cobb, R. C. (1996) Surface deviation modelling of LMT processes – a comparative analysis. In Proceedings of 5th European Conference on *Rapid Prototyping and Manufacturing*, Helsinki, Finland, 4–6 June.
- 12 Fen, X. (1999) Integrated decision support for part fabrication with rapid prototyping and manufacturing systems. PhD thesis, National University of Singapore.
- 13 <http://137.132.146.236/rpdss/benchmark.htm>
- 14 <http://www.raptia.org>
- 15 Rapid Prototyping and Manufacturing Institute –  
<http://rpmi.marc.gatech.edu/build/research.html>
- 16 Rosen, D. W., West, A. P., and Sambu, S. P. (2001) A process planning method for improving build performance in stereolithography. *Computer-Aided Design*, **33**, 65–79.
- 17 Müller, H. and Schimmel, A. (1999) The decision dilemma – assessment and selection of rapid prototyping process chains. In Proceedings of 8th European Conference on *Rapid Prototyping and Manufacturing*, University of Nottingham, UK, pp. 177–192.
- 18 Zhou, J., Herscovici, D., and Chen, C. (1999) Parametric process optimization to improve the accuracy of rapid prototyped stereolithography parts. *International Journal of Machine Tools and Manufacture*, **40**, 1–17.
- 19 Cooper, A. G. (1999) Fabrication of ceramic components using

- mould shape deposition manufacturing. Thesis, Stanford University.
- 20 Halloran, J. (1998) Workshop on materials opportunities in layered manufacturing technologies. Report, University of Michigan, Cosener's House, Abbey Close, Abingdon, Oxfordshire, 22–24 June.
  - 21 Xu, F., Wong, Y. S., Loh, H. T., Fuh, J. Y. H., and Miyazawa, T. (1997) Optimal orientation with variable slicing in stereolithography. *Rapid Prototyping Journal*, **3**(3), 76–88.
  - 22 Cheng, W., Fuh, J. Y. H., Nee, A. Y. C., Wong, Y. S., Loh, H. T., and Miyazawa, T. (1995) Multi-objective optimisation of part-building orientation in stereolithography. *Rapid Prototyping Journal*, **1**(4), 12–23.
  - 23 Dolenc, A. and Makela, I. (1994) Slicing procedures for layered manufacturing techniques. *Computer-Aided Design*, **26**(2), February, 119–126.
  - 24 Zhao, Z. and Laperriere, L. (1998) Adaptive direct slicing of the solid model for rapid prototyping. Universite du Quebec a Trois-Rivieres, Trois-Rivieres, Quebec, Canada.
  - 25 Lee, K. H. and Choi, K. (2000) Generating optimal slice data for layered manufacturing. *International Journal of Advanced Manufacturing Technology*, **16**, 277–284.
  - 26 Allen, S. and Dutta, D. (1995) Determination and evaluation of support structures in layered manufacturing. *Journal of Design and Manufacturing*, **5**, 153–162.
  - 27 Cahlasani, K., Jones, L., and Roscoe, L. (1995) Support generation for fused deposition modelling. In Solid Freeform Fabrication Symposium 1995 (Eds H. L. Marcus *et al.*), University of Texas, Austin, August, pp. 229–241.
  - 28 Marsan, A. L., Kumar, V., Dutta, D., and Pratt, M. J. (1998) *An Assessment of Data Requirements and Data Transfer Formats for Layered Manufacturing NISTIR 6216* (National Institute of Standards and Technology).
  - 29 Kulkarni, P., Marsan, A., and Dutta, D. (2000) A review of process planning techniques in layered manufacturing. *Rapid Prototyping Journal*, **6**(1), 18–35.
  - 30 Smith, R. Standard test-files for benchmarking RP systems. Sherbrook Automotive Limited, Lichfield, UK.
  - 31 Fadel, G. M. and Kirschman, C. (1996) Accuracy issues in CAD to RP translations. *Rapid Prototyping Journal*, **2**(2), 4–17.
  - 32 Gibson, I. and Dongping, S. (1997) Material properties and fabrication parameters in the selective laser sintering process.

*Rapid Prototyping Journal*, 3(4), 129–136.

- 33 Kattethota, G., and Henderson, M. (1998) A design tool to control surface roughness in rapid fabrication. In Proceedings of Solid Freeform Fabrication Symposium 1998, Austin, Texas, pp. 327–334.
- 34 JTEC/WTEC panel report on rapid prototyping in Europe and Japan, Vol. II (1996), Site reports. Baltimore, Maryland, Loyola College. NTIS Report No. PB96-199583.1.
- 35 JTEC/WTEC R. Brown (report author), J. Beaman; Hosts: Kenji Matsuoka, New Product Engineering Department, Olympus Optical Company Ltd (1995–98).
- 36 JTEC/WTEC Attendees: R. Aubin (report author), F. Prinz, C. Uyehara. Site: Kira Corporation, Tomiyoshi Shinden (1996).

## **APPENDIX A3**

### **A3.1 Key considerations in process chains**

#### ***A3.1.1 Orientation (21, 22)***

Orientation is an important criterion in the fabrication of an RP&M component. The orientation of a part as it is built will affect the time needed to build the part, its material properties, surface quality, and the need for support structures. Thus before the part is sliced, its optimal build orientation has to be determined according to criteria important to the designer. With certain RP&M processes, layers that form overhangs or enclosed voids should have a sacrificial support structure beneath them to support the layers as material is added. Such supports are built together with the part, and then detached and discarded. Some process planning systems design the supports after the part has been sliced, while others reverse this order. Finally, scan paths are generated and process parameters are set.

Part orientation and support structure determination are coupled problems, since some orientations of a part will require support structures while others may not. Some RP&M technologies (e.g. SLS and LOM) do not require support structures at all because unused excess material in the build envelope provides the necessary support.

#### ***A3.1.2 Slicing (23–25)***

Slicing is the process of intersecting the model with planes parallel to the platform in order to obtain the contours. Geometric information



processing for RP&M consists of two major steps: slicing the geometric description of the part into layers to generate the contours of the part for each layer, and then generating scan paths for each layer in a manner dependent upon the particular RP&M process.

- Direct slicing. In RP&M, direct slicing refers to slicing of the original CAD model. In this way, the contours are obtained without using an intermediate faceted model such as the STL file. Direct slicing of the solid model keeps the geometric and topological robustness of the original data. Its advantages include greater model accuracy, reduced pre-processing time, elimination of checking and repair routines, and file size reduction.
- Adaptive slicing. The main objective of adaptive slicing is to control the staircase effect by a user-defined tolerance. The other benefit is that the desired surface finish is obtained using the minimal number of layers. In spite of these benefits, its application is presently limited as the building technique is applicable only to those processes that enable variable thickness building. In some cases, the underlying process characteristics are not significantly affected by the layer thickness.

Adaptive direct slicing is a new technology that reaps the benefits of both direct and adaptive slicing (24). Direct slicing generates precise contours for each layer from the solid model and avoids an intermediate representation. Adaptive slicing modifies the layer thickness to take into account the curvature of the surface of the solid model in the vertical direction, to alleviate the staircase effect, and to decrease the number of layers.

### ***A3.1.3 Supported/supportless building (26, 27)***

Rapid prototyping systems are usually bundled with software that allows the automatic creation and editing of the supports. The software will initially generate the supports for all overhang regions based on the default support parameters. After the creation, the support structure can be individually modified, deleted, or added based on the part geometry. The region-by-region editing or customization for support generation is powerful and allows the strengthening of the essential support and also minimizes the building of unnecessary supports. Because of the dynamic viewing of the supports and parts, it is easy and quick to manipulate supports with any change of the support parameters.

#### ***A3.1.4 Support structures (28)***

RP&M support structures are used for a variety of reasons, including supporting overhangs, maintaining stability of the part so that it does not tip over during fabrication, supporting large flat walls, preventing excessive shrinkage, supporting components initially disconnected from the main part, and supporting slanted walls.

Depending on the RP&M process used and its process planning software, support structures are added to the part either before or after slicing. In addition, the types of supports used depend on both the problem to be addressed and the particular RP&M process. In stereolithography, where supports are used to prevent damage to the part caused by application of the material, to prevent curling as the resin hardens, or to support overhangs, many styles of supports are available. However, the FDM process in general uses only thin-walled zigzag supports built up from the base of the modelling envelope or from existing part surfaces.

The use of supports is typically addressed to the following cases: overhangs, floating objects arising during build that later become attached to the part, and part instability during build. In the second and third cases, thin support columns are projected down from the object's surfaces to the base of the part. Overhangs are detected, and appropriate supports generated, on completion of the simulation.

Support structures can be broadly classified into: (a) gusset support, (b) base support, (c) web support, (d) column support, (e) zigzag support (used in FDM), and (f) perimeter and hatch support. The requirements for support structures are generally determined from the geometry of the part, as described by a three-dimensional boundary model or two-dimensional slices, and the maximum draft angle that the selected RP&M process can build without supports. Support structure software generates boundary models of the required supports or adds support contours to a slice representation of the object.

#### ***A3.1.5 Path planning (28, 29)***

It is known by intuition that improved surface accuracy in manufactured parts will result from tracing the contour of each slice rather than simply filling its interior with a series of linear, parallel scan paths. In doing this, the width of the RP&M tool or strand of

material being laid down has to be considered when planning the tool path, by offsetting it inwards from the contour by a distance equal to the tool radius. In manufacturing hollow structures it is efficient to use offset curves rather than raster fills, because the resulting longer paths will reduce the frequency of tool repositioning. The use of offset curves may also reduce the need for support structures, since it enables the construction of an overhang by a sequence of offsets, each of which adheres to the previous one but projects slightly outwards from it.

#### ***A3.1.6 Pre-processing (29–31)***

A pre-processing program prepares the STL file to be built. Research work is in progress to decide on the data transfer format, considering alternatives for the STL files (29–31). Several programs are available, and most allow the user to adjust the size, location, and orientation of the model. Build orientation is important for several reasons. Firstly, properties of rapid prototypes vary from one co-ordinate direction to another. For example, built parts are usually weaker and less accurate in the Z-direction (vertical) than in the X–Y-plane. In addition, part orientation primarily determines the amount of time required to build the model. Placing the shortest dimension in the Z-direction reduces the number of layers, thereby shortening build time.

The pre-processing software slices the STL model into a number of layers from 0.01 to 0.7 mm thick, depending on the build technique. The program may also generate an auxiliary structure to support the model during the part fabrication. Supports are useful for delicate features such as overhangs, internal cavities, and thin-walled sections.

#### ***A3.1.7 Post-processing (9)***

This involves post-treatment after removing the prototype from the machine and detaching supports. Some photosensitive materials need to be fully cured before use. Prototypes may also require minor cleaning and surface treatment. Sanding, sealing, and/or painting the model will improve its appearance and durability.

With SLS, it is not possible to achieve the best quality in appearance with the best mechanical properties. To compensate for this, post-processing may be necessary. In SLS, post-processing includes coating and surface finishing. Coating can improve tensile strength

and surface hardness while surface finishing can improve dimensional and surface precision.

### ***A3.1.8 Process parameters (18)***

When preparing to build RP&M parts, several fabrication parameters have to be taken into consideration. To achieve optimum quality, these parameters are set differently according to the material properties and application requirements. It is therefore important to understand the relationship between the fabrication parameters and the material properties.

There are many process parameters that need to be controlled such as the power, laser scan speed, laser diameter, slice thickness, overlap, etc. There are many other parameters but they are generally process related and vary accordingly.

Some fabrication parameters directly related to part fabrication by the SLS process are described in the following [from fabrication parameters in SLS process by Ian Gibson (32)].

- *Part bed temperature*. The temperature at which the powder in the part cylinder is controlled. Before the laser scanners move, powder in the part cylinder will be heated to part bed temperature.
- *Laser power (P)*. This is power available from the laser beam at part bed surface. This parameter should be set to ensure that the powder at part bed surface would be heated close to  $T_m$  during scanning. The maximum laser power of Sinterstation 2000 is 50 W.
- *Scan size (SS)*. The distance that scanners move in one time step with the laser on. This parameter determines laser beam speed.
- *Beam speed (BS)* =  $SS \times 17.22$  (mm/s).
- *Scan spacing (SCSP)*. The distance between two neighbouring parallel scan vectors. If scan spacing is too great, the cross-section may not be completely sintered. It is related to the laser beam size and energy density.
- *Slice thickness (h)*. The depth that the part piston lowers for each layer, which determines powder thickness of each layer in the part cylinder. A staircase effect is caused by slice thickness. The range of slice thicknesses in the Sinterstation 2000 is 0.07–0.5 mm. The default setting is 0.1 mm.

### **A3.2 Weakness/difficulties**

Many limitations associated with the RP&M process accuracy and repeatability have been improved in recent years with the development of new materials and better process control (33). However, little improvement in the overall surface roughness of components has been seen, without resorting to some form of post-process surface treatment. Current RP&M finishing is undertaken manually. This is actually a significant limitation to the future development of the technology. Constraints also include the support structures and the large horizontal planes.

#### ***A3.2.1 Surface roughness (33)***

Surface finish or surface roughness is a key issue facing RP&M technology. Basically, the manner in which parts are built in RP&M processes by layer addition results in an inferior surface finish, when compared with parts machined on conventional CNC machines. Some of the main causes of surface roughness on RP&M parts are layer thickness, part orientation, the layer build process employed by the particular RP&M system, presence of support structures, etc. (33).

Post-processing the part to get the desired surface finish is a commonly employed technique, but the main advantage of using RP&M for speeding the time to market is reduced by the time delays involved in post-processing. Post-processing may also lead to undesirable loss in dimensional accuracy on the part.

#### ***A3.2.2 Material shrinkage or swelling (9)***

In the early stages of RP&M, the volume changes of the building material during the building process are the most significant limiting factor on part accuracy, and although vast improvements have been made, this is still an important factor. The problem is not so much the volume change, but rather the distortion caused by the non-uniform change, e.g. causing curling of the overhang.

#### ***A3.2.3 Staircase effect (33)***

The staircase effect is intrinsic to the slicing process, and is therefore a common source of dimensional inaccuracy. It has been recognized that in order to improve the surface quality of a part built using RP&M techniques, it is necessary to minimize the staircase effect. Some

RP&M systems allow layers to be added at various angles using a five-axis tool head. This reduces the staircase effect by giving the manufacturer the ability to optimize the build direction for each surface of the part. Most RP&M systems deposit material in only one direction, and reduction of the staircase effect can be achieved by slicing the part at smaller intervals, where possible. This has little effect on the accuracy of the vertical or near-vertical faces, but may significantly improve the surface quality elsewhere. More generally, the slice thickness may be varied non-uniformly during the fabrication process to get the best results, an approach known as adaptive slicing. Adaptive slicing requires two steps: (a) determining the location of peak features and (b) determining the thickness of each slice.

#### ***A3.2.4 Warpage (28)***

Warpage is a problem for the stereolithography process in particular, and means have consequently been sought to optimize scan paths to reduce its effect. Attempts have been made to determine the effect that path planning has on part warpage. A finite element model has been used to predict distortion of a part during build using different deposition strategies. The model was also used to predict how material properties (e.g. strength) vary with the deposition strategy, and how changing build parameters (e.g. ambient temperature) affect residual stress in the part. Parameters such as scan pattern, strand length, strand spacing, overcure, staggering, and retraction are studied and their effects quantified.

**Plastic:** A prime consideration in both Europe and Japan is the development of resins that have improved mechanical properties, better long-term dimensional stability, higher temperature resistance, and lower cost. Better mechanical properties give designers greater latitude in using RP&M models to test functional characteristics (35, 36). It is desirable to achieve properties equivalent to those of ABS. Resin makers are also developing epoxy-based UV curable resins to reduce warping.

**Metal:** There is considerable interest in devising processes that will directly yield metallic components. In processes using metal powder in laser sintering methods, the resulting RP&M products are porous. They can be furnace-sintered to consolidate the objects, but this results in extensive shrinkage and loss of dimensional accuracy. The

alternative is to infiltrate with a lower-melting-point alloy or a high-temperature epoxy resin that does not require furnace sintering. A process to directly make fully dense metal parts by a laser fusion process called laser-generated RP&M (35), uses a milling cutter to trim the walls and surfaces of each layer to improve accuracy and surface finish.

**Ceramics and paper:** A laminated paper (or object) manufacturing (LOM) machine is currently limited to  $\pm 0.1$  mm/25 mm in the horizontal plane ( $x$ - $y$ -direction) and  $\pm 0.3$  mm/25 mm in the vertical build direction ( $z$ -direction). Swelling of the completed part in the  $z$  direction due to humidity is a problem that researchers hope to correct with an improved paper (36).

Benchmarking of materials serves the purpose of identifying with a material that best suits a particular application.

APPENDIX B3

Table B3.1 Principal direct-building materials

Rapid prototyping processes	Prototypes and mould patterns	Short-/medium-run moulds	Long-run moulds
Stereolithography (SLA)	Epoxy resins Acrylate resins		
Solid ground curing (SGC)	Epoxy resins Acrylate resins		
Selective laser sintering (SLS)	Polyamide GF polyamide	Copper polyamide	
Laminated object manufacturing (LOM)	Paper polyester film	Fibre-reinforced composite	Bronze-nickel stainless steel
Fused deposition modelling (FDM)	ABS thermoplastic waxes		

For ease of comparison and understanding the materials are classified into metals and non-metals. Table B2 lists some of the materials that are used in the RP&M industry.

**Table B3.2 Materials**

<b>Material name</b>	<b>Classification</b>	<b>Process used in</b>	<b>Application</b>	<b>Material/prototype properties</b>
SL 5195	Non-metal	SLA	Working prototypes, ACES models, concept models	Low viscosity epoxy resin, prototype with better quality and accuracy
Somos 8110	Non-metal	SLA	Functional applications	Flexible, accurate, high-speed, high-impact-strength epoxy resin
Somos Water Clear 10120 Epoxy	Non-metal	SLA	Applications requiring optical clarity, solid imaging process	Optically clear, rigid, durable epoxy resin
ABS (P400)	Non-metal	FDM	Fully functional, testable prototypes	Durable, high-strength modelling material
ABSi (P500)	Non-metal	FDM	Functional testing	The highest impact strength available
Elastomer (E20)	Non-metal	FDM	Functional, application models	Offers mechanical strength and durability for flexible components
Investment Casting Wax (ICW 06)	Non-metal	FDM	Casting models	Very low ash content and provides superior surface finish
Poly Carbonate	Non-metal	SLS	Functional models	Porous structure
DuraForm GF	Non-metal	SLS	Functional models	Glass-filled polyamide powder, high-durability material; good thermal stability and chemical resistance
Somos 201	Non-metal	SLS	Functional models	Parts are extremely durable and resistant to heat and certain chemicals
SandForm Zr	Non-metal	SLS	Create moulds and cores for sand casting tooling	Pattern removal is quick and easy
CastForm PS	Non-metal	SLS	Foundry models, investment casting	Pattern removal is quick and easy



Table B3.2 continued

Material name	Classification	Process used in	Application	Material/prototype properties
Rapid Steel 2.0	Metal	SLS	Making tooling for injection moulding and die-casting tools; create mould inserts	Greater strength and durability
Copper PA	Metal	SLS	Produces mould inserts	Is durable and thermally conductive, compatible with many common plastics
DuraForm PA	Non-metal	SLS	Functional engineering plastic models	Excellent heat and chemical resistance, machinable
Polystyrene	Non-metal	SLS	Investment casting patterns or visualization models	High-strength rigid parts
Polyamide (nylon)	Non-metal	SLS	Used for parts requiring higher strength and/or toughness	Higher strength and/or toughness
LaserForm ST-100	Metal	SLS	Moulding of most thermoplastics	High thermal conductivity for short cycle times; good accuracy, especially when machined in tight tolerance area

# Chapter 4

## Decision support and system selection for RP

**D W Rosen<sup>1</sup> and I Gibson<sup>2</sup>**

<sup>1</sup>The George W. Woodruff School of Mechanical Engineering,  
Georgia Institute of Technology

<sup>2</sup>Department of Mechanical Engineering, The University of Hong  
Kong

### 4.1 INTRODUCTION

In his Rapid Prototyping and Tooling State of the Industry Report for the year 2000, Terry Wohlers (1) estimates that there are over 6500 RP machines in use worldwide, producing over 2.3 million parts per year. Growth in the RP market has been relatively constant over the last few years at around 20 per cent per annum. Around the world, there are over 20 system manufacturers selling around 40 different models of RP machine. Around 10 new models came on to the market in the year 2000. This points to an increasingly important market that is becoming more complex in terms of variety of machines and choices associated with those machines (for example the range of materials, speed options, accuracy options, etc.).

The initial purpose of RP technology was to create parts as a means of visual and tactile communication. According to Wohlers and Grimm, RP parts are used for a number of purposes, including:

- Communication of product design concept
- Engineering changes
- Support for new product proposals

- Verification of CAD databases
- Styling and ergonomic studies
- Functional testing
- Prototypes and prototype tooling
- Metal casting patterns
- Early notification of tooling changes
- Requests for quotations
- Rapid manufacturing

The focus is on plastic products: either to model the products themselves or to develop the tooling to mass-produce them. However, this is a vast market and there is plenty of room for growth. Rapid prototyping technology, like all materials processing, is constrained by material properties, speed, cost, and accuracy. The performance constraints of materials and accuracy are well below conventional manufacturing technology (e.g. injection moulding machinery). Speed and cost, in terms of time to market, are where RP technology contributes.

If this market growth continues, then there is going to be increasing demand for software that allows us to make decisions on what machines to use, why we should use them, and what we should expect from them. In particular, software systems can help in the decision-making process for capital investment of new technology, providing accurate estimates of cost and time for quoting purposes, and assistance in process planning.

This chapter deals with three typical problems involving RP that may benefit from decision support:

1. Quotation support. Given a part, what machine and material should I use to build?
2. Capital investment support. Given a design and industrial profile, what is the best machine that I can buy to fulfil my requirements?
3. Process planning support. Given a part and a machine, how do I set it up to work in the most efficient manner alongside my other operations and existing tasks?

Examples of systems designed to fulfil the first two problems are described in detail. The third problem is much more difficult and is discussed only.

## 4.2 SELECTION METHODS FOR A PART

Nominally, the problem to be addressed here is when a designer wants a part to be prototyped; an RP process must be selected to fabricate it. Also, a suitable material must be selected. This problem can be generalized in several manners to cover all the parts in an assembly, or to address process chains required to fabricate a part. More explicitly, the problem being addressed will be termed material process selection (MPS) and is defined as:

**Problem MPS:** Given a CAD model of a part and prototyping requirements, select a suitable material and RP process for that part to best meet all of the prototyping requirements.

The term ‘prototyping requirements’ refers to all of the designer’s requirements including, for example, the prototype’s purpose, which tests will be performed on the prototype, preferences for materials, limits on cost and/or delivery time, etc.

The other problem type to be addressed in this section is the material process chain selection (MPCS) problem, which can be defined as:

**Problem MPCS:** Given a CAD model of a part and prototyping requirements, select a suitable material and sequence (chain) of processes for that part to best meet all of the prototyping requirements.

In this case, a process chain means a sequence of operations or processes required to fabricate the part. A good example of this is the use of rapid tooling to enable the injection moulding of prototypes. An RP process may be used to fabricate injection-moulding tools directly, or may be used to fabricate a pattern from which the tool is produced. In the case of problem MPCS, the generation of feasible process chains is a difficult problem.

The selection problems being addressed in this section will be divided into two related subproblems. First, it is necessary to generate feasible alternatives, which, in this case, include materials and processes. Second, given those feasible alternatives, a quantification process is applied that results in a rank-ordered list of alternatives. The first subproblem is referred to as ‘determining feasibility,’ while the second is called simply ‘selection’.

#### 4.2.1 Approaches to determining feasibility

The problem of identifying suitable materials and RP machines with which to fabricate a part is surprisingly complex. As noted above, there are many possible applications for an RP part. For each application, one should consider the suitability of available materials, fabrication cost and time, surface finish and accuracy requirements, part size, feature size, mechanical properties, resistance to chemicals, and other application-specific considerations. To complicate matters, the number and capability of commercial materials and machines continues to increase. Therefore, in order to solve problem MPS and MPCS, one must navigate the materials and machine, comparing one's needs to their capabilities, while ensuring that the most up-to-date information is available.

To date, most approaches to determining feasibility have taken a knowledge-based approach in order to deal with the qualitative information related to RP process capability. One of the better-developed approaches was presented by Deglin and Bernard (2). They presented a system for the generation, selection, and process planning of rapid manufacturing processes. The problem as they defined it was: 'To propose, from a detailed functional specification, different alternatives of rapid manufacturing processes, which can be ordered and optimized when considering a combination of different specification criteria (cost, quality, delay, aspect, material, etc.)'. Their knowledge-based approach was capable of two reasoning methods: case-based and the bottom-up generation of processes. In their view, the advantage of the case-based approach is the foundation of validated case information. A disadvantage is that many cases are needed in order to provide the desired coverage of RP problems and applications. On the other hand, the advantage of the generative approach is the ability to generate process chains that would not otherwise be found in a limited case base. Both reasoning methods have their role in the solution of MPS and MPCS problems. Their system was developed on the KADVISER platform and utilizes a relational database system with extensive material, machine, and application information.

Another group at the National University of Singapore (NUS) has developed an RP decision system that is integrated with a database system (3). Their databases and benchmarking activities were reported in Chapters 2 and 3 of this book. Their selection system is capable of

identifying feasible material/machine combinations, estimating manufacturing cost and time, and determining optimal part orientations. From the feasible material/machines, the user can then select the most suitable combination. Their approach to determining feasible materials and processes is broadly similar to the work of Deglin and Bernard. The NUS group utilized five databases, each organized in a hierarchical, object-oriented manner: three general databases (materials, machines, and applications) and two part-specific databases (geometric information and model specifications). Please see Chapter 2 for more information on these databases.

Several web-based RP selection systems are available. One was recently developed at the Helsinki University of Technology (see <http://ltk.hut.fi/RP-Selector>). Through a series of questions, the selector acquires information about the part accuracy, layer thickness, geometric features, material, and application requirements. The user chooses one of 4–5 options for each question. Additionally, the user specifies preferences for each requirement using a five-element scale from insignificant to average to important. When all 10–12 questions are answered, the user receives a set of recommended RP machines that best satisfy their requirements.

#### **4.2.2 Approaches to selection**

As stated earlier, there have been a number of approaches taken to support the selection of RP processes for a part. For the purposes of illustration, the approach based on decision theory, particularly the ‘decision support problem technique’ (DSP), will be presented.

##### ***4.2.2.1 Decision support problem technique***

Mistree, Allen, and co-workers (4–6) have been developing the DSPs technique over the last 20 years. Recently, they have applied this work to RP selection and related problems. The advantages of selection DSPs, compared with other decision formulations, are that they provide a means for mathematically modelling design decisions involving multiple objectives and supporting human judgment in designing systems (7, 8). Although all of the information needed to model a system comprehensively may not be available, especially in the early stages of design, DSPs facilitate the search for superior or satisfying solutions (9). The formulation and solution of DSPs facilitate several types of decisions, including:

*Selection*: the indication of preference, based on multiple attributes, for one among several alternatives (10);

*Compromise*: the improvement of an alternative through modification (5, 6);

*Coupled and hierarchical*: decisions that are linked together, such as selection–selection, compromise–compromise, or selection–compromise (4).

While the basic advantages of using DSPs of any type lie in providing context and structure for engineering problems, regardless of complexity, they also facilitate the recording of viewpoints associated with these decisions, for completeness and future reference, and evaluation of results through post-solution sensitivity analysis. The standard selection decision support problem (sDSP) has been applied to many engineering problems and has recently been applied to RP selection (11). The word formulation of the standard sDSP is given in Fig. 4.1. Note that the decision *options* for RP selection are feasible material–process combinations. Expectations are determined by rating the options against the *attributes*. Preferences are modelled using simple *importance* values. Rank ordering of options is determined using a weighted-sum expression of importance and attribute ratings. An extension to include utility theory has recently been accomplished, as described next.

#### 4.2.2.2 Utility theory based selection

By complementing the standard selection DSP with utility theory, an axiomatic basis is provided for accurately reflecting the preferences of a designer for trade-offs and uncertainty associated with multiple attributes. Additional background on decision theory is given below in Box 1. The new utility theory based selection DSP is termed the u-sDSP.

##### Selection DSP

**Given:** Set of feasible alternatives.

**Identify:** Key attributes that influence the selection of alternatives.  
Relative importance of attributes.

**Rate:** Alternatives with respect to each attribute.

**Rank:** Order the alternatives in terms of preference.

**Fig. 4.1 Selection DSP word formulation**

The notion of uncertainty with respect to attributes is explicitly modelled and incorporated within the u-sDSP and care must be taken to identify not only attributes but also any uncertainty associated with them. In order to provide a rigorous, preference-consistent alternative to the traditional merit function for considering alternatives with uncertain attribute values, we use expected utility [e.g. Keeney and Raiffa (12)]. This requires the satisfaction of a set of axioms such as those proposed by von Neumann and Morgenstern (13), Luce and Raiffa (14), or Savage (15), describing the preferences of rational individuals. Once satisfied, there then exists a utility function with the desirable property of assigning numerical utilities to all possible consequences such that the best course of action for the individual is indicated by the highest expected utility.

Utility theory has been employed in several engineering design contexts for evaluating alternatives (7, 8, 16, 17). Here the focus is on updating the selection DSP to provide additional rigour, especially in the context of uncertainty, while preserving the role of a designer in the decision-making process. Much of this material is based on the paper by Fernandez *et al.* (18). The word formulation of the u-sDSP is illustrated in Fig. 4.2.

The u-sDSP provides structure and support for using human judgment in engineering decisions involving multiple attributes and facilitates the explicit consideration of a decision maker's preferences in the context of risk and uncertainty. The method is mathematically rigorous – based on a set of clearly defined axioms, rather than heuristics – and provides consistently reliable results when employed in a proper

#### Utility Theory Selection DSP

**Given:** Set of feasible alternatives.

**Identify:** Key attributes influencing selection.

Uncertainties associated with each attribute.

**Assess:** Decision maker's utility with respect to each attribute and with respect to combinations of attributes.

**Evaluate:** Each alternative using the decision maker's utility functions.

**Rank:** Order the alternatives based on expected utility.

**Fig. 4.2 u-sDSP word formulation**



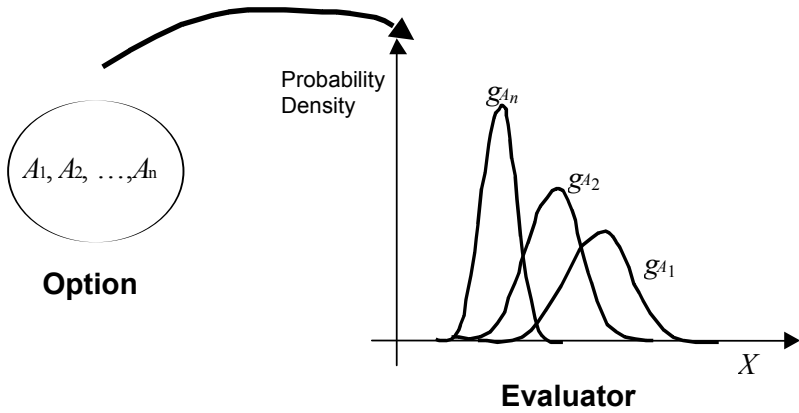
context. It is also preference consistent because it facilitates the explicit and accurate incorporation of a designer's preferences for multiple attributes as well as trade-offs and uncertainty related to these attributes. Overall, it facilitates rational decision-making (with rational defined as being consistent with a designer's preferences). This capability to model preferences quantitatively becomes especially significant when dealing with a large number of objectives, for which simultaneous *ad hoc* consideration becomes close to impossible and treatment of trade-offs and uncertainty becomes crucial. Finally, as demonstrated in the context of rapid prototyping resource selection, the method is useful: it is implementable with reasonable effort and enhances a designer's ability to make complex selection decisions. Each element of the u-sDSP will be described briefly.

### Box 1: Decision theory

Decision theory has a rich history, evolving in the 1940s and 1950s from the field of economics (12). Although there are many approaches taken in the decision theory field, the focus in this chapter will be only on the utility theory approach. Broadly speaking, there are three elements of any decision (17):

- Options – the items from which the decision maker is selecting,
- Expectations – of possible outcomes for each option, and
- Preferences – how the decision maker values each outcome.

Assume that the set of decision options is denoted  $A = \{A_1, A_2, \dots, A_n\}$ . In engineering applications, one can think of outcomes as the performance of the options as measured by a set of evaluation criteria. More specifically, in RP selection, an outcome might consist of the time, cost, and surface finish of a part built using a certain RP process, while the RP process itself is the option. Expectations of outcomes are modelled as functions of the options,  $X = g(A)$ , and typically have uncertainties associated with them, modelled as probability distributions. Figure B1 shows the relationship between options and expectations assuming only one decision criterion (only one type of expectation, e.g. cost).



**Fig. B1 Mapping options to expectations in unidimensional utility theory**

Preferences model the importance assigned to outcomes by the decision maker. For example, a designer may prefer low cost and short turnaround times for a concept model, while being willing to accept poor surface finish. In utility theory, preferences are modelled as utility functions on the expectations. Mathematically, let option  $A_i$  result in outcome  $x_i \in X$  with probability  $p_i$ , if outcomes are discrete. Otherwise, expectations on outcomes are modelled using probability density functions,  $f_i = f_i(x_i)$ . Utility is denoted  $u(x)$ . Expectations are then modelled as expected utility as shown in equation (B1).

$$E[u(x)] = \sum p_i u(x_i) \quad \text{for discrete outcomes}$$

$$E[u(x)] = \int u(x) g(x) \quad \text{for continuous outcomes} \quad (\text{B1})$$

This leads to the primary decision rule of utility theory:

*Select the option whose outcome has the largest expected utility.*

Note that expected utility is a probabilistic quantity, not a certain quantity, so there is always risk inherent in these decisions.

The objective of utility measures is that they can be used to rank order options. Given two options  $A$  and  $B$  with utilities  $u_A$  and  $u_B$ , respectively, the decision maker may prefer  $A$  to  $B$ . This is denoted  $A \succ B$ , and it must be true that  $u_A > u_B$ , if  $u_i$ 's are truly utilities. Similarly, if the decision maker is indifferent to  $A$  and  $B$  ( $A \sim B$ ), then  $u_A = u_B$ . Generalizing to  $n$  options, the application of utilities can rank order the options in terms of preference.

Two key concepts underlie the application of utility functions in practice. One is the concept of *indifference*. Consider two RP attributes (examples of outcome types) such as cost (call it  $A$ ) and surface finish ( $B$ ). If the decision maker is indifferent between cost and surface finish, then  $A \sim B$ . However, this indifference may only apply for certain levels of  $A$  and  $B$ . In fact, it is typical that the preference relationship between  $A$  and  $B$  follows an indifference curve, the locus of points joining  $A$  and  $B$ . Think of plotting such a curve in  $A$ – $B$  space. For example, assume that the part cost is \$200. The value of surface finish at which the decision maker is indifferent may be 100  $\mu\text{m}$ . However, if the cost is \$300, the indifference value for surface finish is likely to be more demanding, for example 70  $\mu\text{m}$ . In general, one can plot indifference curves (or hypersurfaces) between any collection of attributes.

The other key concept is that of *marginal rate of substitution*. Given attributes  $X$  and  $Y$ , if  $Y$  is increased by  $\Delta$  units, how much does  $X$  have to decrease in order for the decision maker to be indifferent? The answer probably depends upon the particular point at which the question is asked. The marginal rate of substitution is the slope of the indifference curve at the particular point  $(x, y)$ . That is, rate =  $\Delta Y / \Delta X$  at  $(x, y)$ . By applying these concepts carefully, several methods of eliciting utilities from decision makers are possible (8, 16).

Given that utilities for individual attributes can be captured from decision makers, the next challenge is to combine them into a single measure of utility that can be applied to all outcomes of interest. That is, a scalar measure is needed to rank order the outcomes. A variety of additive and multiplicative forms for multi-attribute utility functions have been developed. Each should be applied under narrowly defined assumptions about the decision maker's preference structure. For example, the simplest additive form can

only be applied when the additive independence condition holds, meaning that the levels of those outcomes do not influence preferences among outcomes and that no interactions exist among the preferences of the outcomes. Although this is a rather restrictive condition, it is often reasonable in engineering situations. The additive multi-attribute utility function has the following form:

$$u(x) = \sum_{i=1}^n k_i u(x_i) \quad (\text{B2})$$

where the  $k_i$  are scaling constants ( $\sum k_i = 1$ ) that act like weights in typical weighted-sum expressions. These scaling constants can be evaluated by querying the decision maker with easily defined questions about his/her preferences.

**Given:** The alternatives are simply the set of feasible RP process and material pairs under consideration. For example, if an SLA-5000 machine is available, then several of the feasible process-material pairs may be SLA5000-SL7510, SLA5000-SL5540, but not SLA5000-SL5170, because SL5170 is a material only for the SLA-250 machine.

**Identify:** The first set of information to be identified is the attributes that describe the alternatives' characteristics of interest. For RP selections, these attributes will probably include cost, time, accuracy, surface finish, and other part properties that are related to the prototype's purpose. Secondly, the uncertainties associated with each attribute must be specified. These may be specified using uniform or other probability distributions, depending upon the information available about the attribute.

**Assess:** In this step, the decision maker specifies his/her preferences regarding the attributes. As a result, a mathematical model of preference is captured, based in utility theory. First, a utility function is captured for each attribute (e.g. without considering other attributes, specify the importance of *prototype cost* for various values of cost). These utility functions are the  $u(x)$  in Box 1. Then, utility models are derived for each combination of attributes (scaling constants,  $k_i$ ). In this way, the decision maker explicitly specifies the trade-offs inherent among the attributes.

**Evaluate:** First, each alternative is rated against each attribute. For example, cost, time, and surface finish attributes would be determined for each RP material–process alternative under consideration. Then, the utility functions for those attributes are applied. Finally, the multi-attribute utility function, equation (B2), is computed for each alternative.

**Rank:** The expected utility of each alternative is computed using equation (B1). This enables the alternatives to be rank ordered in terms of preference.

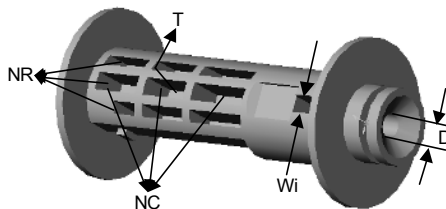
#### 4.2.2.3 *RP selection example*

The manufacturer of a single-use camera requires several prototypes of the film spool shown in Fig. 4.3. For this problem, the manufacturer requires functional prototypes for use in physically testing prototype cameras, particularly for winding and unwinding film onto the spool. As such, the primary objectives in decreasing order of importance are the following:

- Functional product validation, particularly with respect to strength and endurance
- Determining the closeness of fit/tolerance of the interfacing components
- Obtaining a feel for the product
- Visual and physical confirmation of three-dimensional interface integrity

Thus, the following properties are of primary interest with respect to these objectives. Production parts will be moulded in polystyrene.

- Tensile Strength
- Flexural Modulus
- Young's Modulus
- Detail Capability
- Flexural Strength
- Accuracy



**Fig. 4.3** Camera spool example

The alternative materials and processes are given in Table 4.1. These were identified as feasible since they are available in the Georgia Tech Rapid Prototyping and Manufacturing Institute.

**Table 4.1 Feasible materials and processes**

<b>SOMOS 7110</b> – General-purpose photopolymer that produces rigid, highly accurate parts. Compatible with the SLA-250.	<b>SLA-250</b> – Stereolithography machine with $250 \times 250 \times 250$ mm build volume and HeCd laser.
<b>SOMOS 8120</b> – Photopolymer that produces accurate, flexible parts with high impact strength. Intended to mimic the mechanical properties of polyethylene. For the SLA-3500/5000/7000.	<b>SLA-3500</b> – Stereolithography machine with $350 \times 350 \times 400$ mm build volume and solid state laser.
<b>SL 7510</b> – General-purpose photopolymer for the SLA-3500/5000/7000.	<b>FDM 1650</b> – Fused deposition modelling machine with a build volume of about $250 \times 250 \times 300$ mm.
<b>P400</b> – Durable, high-strength ABS modelling material available for FDM machines.	<b>ACTUA 2100</b> – Multi-jet modelling machine used for fabrication of concept models.
<b>TJ 65</b> – Thermojet material that is durable enough for intended use in concept models, although not intended for use in functional testing or in applications that overly stress the model.	

Decision attributes must be quantified by specifying scales and units and are shown in Table 4.2. The decision maker also needs to specify probability distributions for each attribute. Also, these distributions will probably differ for different alternatives. Uniform distributions are specified for this problem, as shown in Table 4.3.

For the material property attributes, the decision maker wants the prototype's properties to match those of the production part. The prototype's properties may deviate either below or above those of the production part; therefore two-sided preference models are utilized (called 'Target' monotonicity in Table 4.4). Hence, the left-hand side is modelled using a utility function that is increasing, while the right-

hand side function is decreasing. Since detail capability and accuracy should be minimized, their utility functions are monotonically decreasing. Table 4.4 contains the preference information for each attribute. Utility functions for each attribute can then be computed, as shown in Table 4.5.

The next step is to capture additional preference information from the decision maker in order to determine the scaling constants,  $k_i$ , from equation (B2). The scaling constants used for combining the utility functions for individual attributes into one multi-attribute utility function are assessed by determining levels of an attribute required to warrant choosing an improvement in that attribute over a loss of performance with respect to all other attributes as described previously. For this problem, the decision maker has specified the value of an attribute, denoted by  $TS_7$ , for which he/she is indifferent between the following two options.

**Option A:** Having the attribute in question at a level of utility = 0.45 and all remaining attributes at a slightly higher level, utility = 0.55.

$$(TS_0, YM_{0.55}, FS_{0.55}, FM_{0.55}, DC_{0.55}, A_{0.55})$$

**Option B:** Having the attribute in question at a level specified by the decision maker, denoted here by  $TS_7$ , and all remaining attributes at a less preferred level (utility = 0.45).

**Table 4.2 Attributes and their units, descriptions, scales, and values**

Attribute	Acronym	Description	Units	Scale	Lower un-acceptable	Ideal	Upper un-acceptable
Tensile strength	TS	The strength of the material under tension	MPa	Ratio	22	37	52
Young's modulus	YM	The modulus of elasticity of the material	MPa	Ratio	2700	3200	3700
Flexural strength	FS	The strength of the material under bending	MPa	Ratio	45	67	89
Flexural modulus	FM	The modulus of flexural stiffness of the material	MPa	Ratio	2500	3000	3500
Detail capability	DC	A measure of a process's resolution (i.e. what level of detail a machine is capable of reproducing – smallest possible feature size)	mm	Ratio	0.8	0.4	—
Accuracy	A	The capability of a machine to maintain the dimensional requirements posed when using a specified material	Qualitative	Interval	0.1	0.02	—

$$(TS_7, YM_{0.45}, FS_{0.45}, FM_{0.45}, DC_{0.45}, A_{0.45})$$

For example,  $YM_{0.55}$  is the value of Young's modulus corresponding to a utility of 0.55 according to its single attribute utility function. Resulting  $k_i$  values are shown in the last column of Table 4.6.

Having captured the decision maker's overall preference structure, the decision maker's expected utility can be calculated for each alternative.

**Table 4.3 Probability distributions of alternatives**

		Attributes								
Alternatives		Tensile strength			Young's modulus			Flexural strength		
Process	Material	Type of distribution	Lower bound/mean	Upper bound/variance	Type of distribution	Lower bound/mean	Upper bound/variance	Type of distribution	Lower bound/mean	Upper bound/variance
SLA250	DSM7110	Uniform	44	69	Uniform	1758	2413	Uniform	59	110
SLA3500	SL7510	Uniform	42.3	55.46	Uniform	1877	2869	Uniform	78	96
SLA3500	DSM8120	Uniform	23	29	Uniform	633	773	Uniform	23	29
FDM1650	P400	Uniform	31	37	Uniform	2234	2730	Uniform	58	72
MJM2100	TJ75	Uniform	9	11	Uniform	90	110	Uniform	9	11

		Attributes								
Alternatives		Flexural modulus			Detail capacity			Accuracy		
Process	Material	Type of distribution	Lower bound/mean	Upper bound/variance	Type of distribution	Lower bound/mean	Upper bound/variance	Type of distribution	Lower bound/mean	Upper bound/variance
SLA250	DSM7110	Uniform	1710	2668	Uniform	0.45	0.55	Uniform	0.04	0.05
SLA3500	SL7510	Uniform	2374	2902	Uniform	0.45	0.55	Uniform	0.04	0.05
SLA3500	DSM8120	Uniform	621	759	Uniform	0.45	0.55	Uniform	0.04	0.05
FDM1650	P400	Uniform	2358	2882	Uniform	0.45	0.55	Uniform	0.121	0.135
MJM2100	TJ75	Uniform	0	110	Uniform	0.67	0.83	Uniform	0.121	0.135

**Table 4.4 Monotonicity, attitude towards risk, and preference levels of attributes**

Attribute	Monotonicity	Attitude towards risk	Left-hand-side utility					Right-hand-side utility				
			0	0.25	0.5	0.75	1	0.75	0.5	0.25	0	
Tensile strength	Target	Averse	22.00	23.70	26.30	30.60	37.00	42.30	46.20	49.50	52.00	
Young's modulus	Target	Averse	2700.00	2756.00	2843.00	2986.00	3200.00	3377.00	3506.00	3616.00	3700.00	
Flexural strength	Target	Averse	45.00	47.50	51.30	57.60	67.00	74.80	80.50	85.30	89.00	
Flexural modulus	Target	Averse	2500.00	2556.00	2642.00	2786.00	3000.00	3177.00	3306.00	3416.00	3500.00	
Detail capability	Decreasing	Averse	0.80	0.73	0.64	0.54	0.40	—	—	—	—	
Accuracy	Decreasing	Averse	0.100	0.087	0.069	0.048	0.020	—	—	—	—	



**Table 4.5** Coefficients of attribute utility functions of the general form  $y = a + bx + ce^{dx}$

Attribute	Left-hand-side				Right-hand-side			
	a	b	c	d	a	b	c	d
Tensile strength	1.0218	0	-33.6690	-0.1598	2.0175	0	-0.1810	0.0462
Young's modulus	1.0221	0	-429526.4240	-0.0048	2.0175	0	-0.0118	0.0014
Flexural strength	1.0216	0	-135.4810	-0.1091	2.0180	0	-0.1210	0.0315
Flexural modulus	1.0220	0	-170436.3600	-0.0048	2.0175	0	-0.0156	0.0014
Detail Capability	2.0175	0	-0.4994	1.7357	0	0	0	0
Accuracy	2.0160	0	-0.8413	8.6390	0	0	0	0

These expected utilities take into consideration not only the relative importance of certain attributes within the overall preference structure but also the uncertainty associated with each attribute for each alternative. The results of these calculations are provided in Table 4.7.

The most preferred alternative is the one with the highest expected utility. As evidenced by the results, the final material–process combination most suited to meeting the needs of our decision maker is using P400 ABS material in the FDM1650 machine. Considering the alternatives, this result seems reasonable. The negative expected utility values indicate that there is not a good match between the alternative’s capabilities and the problem requirements. Hence, not only is the P400/FDM1650 alternative the most favourable, it is really the only feasible alternative.

**Table 4.6** Scaling factor values

Attribute	k-Values
Tensile strength	0.149901269
Young's modulus	0.150002349
Flexural strength	0.157980133
Flexural modulus	0.171275645
Detail capability	0.184126699
Accuracy	0.186713905

**Table 4.7** Selection results

Alternatives		Expected utility
Process	Material	
FDM1650	P400	0.18973
SLA3500	SL7510	-1.059082
SLA250	DSM7110	-5.320072
SLA3500	DSM8120	-331.96516
MJM2100	TJ75	-57912.53518

### 4.3 A RAPID PROTOTYPING SYSTEM SELECTION GUIDE

The rationale behind creating this system comes from the fact that different RP systems are focused on slightly different markets. For example, there are large, expensive machines that can fabricate parts using a variety of materials with relatively high accuracy and/or material properties and with the ability to fine-tune the systems to meet specific needs. In contrast there are cheaper systems, which are designed to have minimal set-up and to produce parts of acceptable quality in a predictable and reliable manner. In this latter case, parts may not have high accuracy, material strength, or flexibility of use.

Different users will require different things from an RP machine. Machines vary in terms of cost, size, range of materials, accuracy of part, time of build, etc. It is not surprising to know that the more expensive machines provide the wider range of options and therefore it is important for someone looking to buy a new machine to be able to understand the costs versus the benefits so that it is possible to choose the best machine to suit their needs.

Approaching a manufacturer or distributor of RP equipment is one way to get information concerning the specification of their machine. Such companies are obviously biased towards their own product and therefore it is going to be difficult to obtain truly objective opinions. Conventions and exhibitions are a good way to make comparisons, but these do not easily identify the usability of machines. Contacting existing users is sometimes difficult, but they can give very honest opinions. This, however, is a time-consuming task if you are looking for opinions on a large range of machines. This approach works best if you are already equipped with background information concerning your proposed use of the technology.

This section describes some initial investigations into creating a web-based software tool to assist engineers and managers who are new to the RP industry and technology. There were a number of considerations when trying to set up this system:

- The information provided should be unbiased wherever possible.
- The system should provide support and advice rather than just a quantified result.

- The system should provide an introduction to RP to equip the user with background knowledge as well as advice on different RP technologies.
- A range of options should be given to the user in order to adjust his requirements and show how changes in requirements may affect the decision.
- The system should be linked to a comprehensive and up-to-date database of systems.
- Once the search process has been completed, the system should give guidance on where to look next for additional information.

The process of accessing the system should be as beneficial to the user as the answers it gives. However, this is not as easy a task as one might first envisage. If it were possible to decouple the attributes of the system from the user specification, then it would be a relatively simple task to select one machine against another. To illustrate that this is not always possible, consider the following scenarios.

1. In an SLS machine, warm-up and cool down are important stages during the build cycle that do not directly involve parts being fabricated. This means that large parts do not take proportionally longer times to build compared with smaller ones. Large builds are more efficient than small ones. In SLA and FDM machines there is a much stronger correlation between part size and build time. Small parts would therefore take less time on an SLA or FDM machine than when using SLS, if considered in isolation. Many users, however, batch process their builds and the ability to vertically stack parts in an SLS machine makes it generally possible to utilize the available space more efficiently. The warm-up and cool down overheads are less important for larger builds and the time per layer is generally quicker than most SLA and FDM machines. As a result of this discussion it is not easy to see which machine would be quicker without carefully analysing the entire process plan for using a new machine.
2. Generally, it costs less to buy a Thermojet machine compared with a ZCorp machine. There are technical differences between these machines that make them suitable for different potential applications. However, because they are in a similar price bracket, they are often compared for similar applications. Thermojet machines use a cartridge-based material delivery system that requires a complete replacement of the cartridge when empty. This makes material use much more expensive when compared with the

ZCorp machine. For occasional use it is therefore perhaps better to use a Thermojet machine when all factors are equal. The more parts you build, the more cost-effective the ZCorp machine becomes.

This, plus many other similar scenarios, indicate that an expert system is likely to form part of the selection software. The purpose is to attempt to embody the expertise resulting from extensive use of RP technology into a software package that can assist the user in overcoming at least some of the learning curve quickly and in a single stage.

#### **4.3.1 Expert system selection**

A number of software systems were evaluated for this project. Since the original intention was to create a web-based tool, a suitable web-publishing package was chosen. An undergraduate student carried out the work as part of his final-year studies in the form of a final-year project. Students are taught to use Microsoft FrontPage as part of their undergraduate study and so this was chosen as the basis for creating the overall format. FrontPage is used for creating HTML files, which is not a suitable platform for creating an expert system. The software systems available that were considered suitable were Java 2 SDK, CLIPS 6.10, KAPPA-PC 2.01, and JESS 4.4. The first package is a software development kit (SDK) while the remaining ones are expert system shells. The choice therefore was whether to use an existing shell or to develop an expert system from scratch. Table 4.8 shows a comparison of the different software systems.

A further objective of the system was that it should be compatible with a separate database that may even be running on a remote site. The Java 2 SDK was chosen to develop the web-based tool. Despite the fact that it is not an expert system shell, it does have a number of features that facilitate expert system creation. The Java Bean class library enables file transfer between different software packages, like databases. The KAPPA-PC system is not web-enabled and so it may be difficult to access the system remotely. The Java 2 SDK is the most portable of the available systems, capable of being developed in an applet format for embedding into web pages and enabling public access. The decision was not made on whether to use this facility or to restrict access. The value of the system is most probably in the data and separating out this from the searching system may be sensible from a commercial point of view.

**Table 4.8 Comparisons of the software packages for selection program development**

Software packages comparisons					
Features		CLIPS 6.0	KAPPA-PC 2.01	JESS 4.4	Java 2 SDK
Expert system shell		✓	✓	✓	✗
Running platform		Microsoft Windows 98	Microsoft Windows 3.1	Java Runtime Environment	Cross-platform
IDE		✓	✓	✗	✗
Debugger		✓	✓	✗	✓
Object oriented programming		✓	✓	✗	✓
Compiled code	GUI application	✗	✓	✗ (*)	✓
	Format	Clips script	Kappa file	Jess script	Applet, JRE Application
	Running platform	Windows 98	Windows 3.1	JRE	JRE, WWW Browser
	Distribution	Difficult	Difficult	Easy	Easy
Complete manual		✓	✓	✗	✓
Accessing database		✗	✓	✗	✓
Software update		✓	✗	✓	✓

Note: IDE = Integrated Development Environment; GUI = Graphical Users Interface; JRE = Java Runtime Environment; (\*) = GUI can be built by the use of Java 2 SDK.

### 4.3.2 Rules and weighting system

With the software chosen, the next stage was to generate the environment and the rules to go into the expert system. The RP selection process was divided into rules associated with part application, characteristics of the prototype, and the working environment.

From the point of view of part application, first of all a general match was chosen in relation to the industrial sector. Some machines focus on specific industrial sectors (e.g. jewellery, medical industry, etc.) and so particular machines may be favoured for a specific industry. Then, the application is divided into the different purposes RP parts are commonly used for. These attributes include concept modelling, functional testing, casting, short-run tooling, etc. Questions were also asked to establish the likely frequency each of these application areas is likely to be put to use. In this way an RP machine that does not produce good functional testing parts would score low for this category, but if not many functional parts are made, the impact of this score is reduced. The application also asks a question relating to how quickly a model is required. As mentioned before, this is a very subjective issue. The most likely way to understand the impact of this question is to look at how the decision-making process is affected by multiple use of the software tool. Asking for a very fast machine may bias the choice in favour of the ZCorp process (obviously dependent on other factors being chosen as well). By reducing the speed threshold, other machines will start to show up.

The characteristics of the prototype relate mainly to physical properties. These include part size, accuracy measures (resolution, distortion, surface roughness, etc.), resistance to heat, water, and chemicals, and strength measurements. This is a complex range of properties since some machines can process a number of different materials. One material may not provide all the required characteristics and so the potential user should be alerted to this fact. Parts larger than the working volume can still be made in sections and assembled or glued together. Similarly, parts can be post-processed to improve their physical attributes. The highest score would be given to a process that has a single material that can meet all the requirements of the user. Other processes and materials will score well if they can meet the requirements without significant post-processing. The user can, however, be alerted to the fact that a process can still be suitable for a limited range of applications. If an unusual request is made for a higher performing product, it is still possible to use the process provided a little care is taken during the post-processing stages.

The working environment deals with compatibility between the machine and the workplace. This relates to issues such as power rating, size of machine, and whether there are any specific requirements for

installation (fume extraction, compressed air or gas input, etc.). Some machines are designed for design office environments while others are more commonly placed on the shop floor. The user of the software tool should be made aware of this, particularly if he/she has limited facilities for installation.

As can be seen from above, the questions asked can be both ambiguous and/or misleading. Users should be provided with additional information regarding the purpose of each question and the problems associated with the answer. The web-based system is therefore envisaged as a learning tool rather than a decision-making tool. Even with a relatively limited range of machine options such as in RP, the chances of obtaining a biased answer is very high. By equipping the user with the reasoning process, this bias should not be so much of a concern. The other issue relates to the local market and environment. Not all machines are available all over the world; plus local issues like shipping, materials supply, customer support, and negotiations on individual machine price can also be significant to the decision-making process. A web-based tool that is available around the world can again only highlight these issues rather than factor them into any equation. This explains why there are no specific questions relating to the cost of the machine in this tool.

#### **4.3.3 Example web pages and search process**

Once the user has accessed the home page for the decision support system (Fig. 4.4a), the user is first given the opportunity to understand RP and the associated technology. The user can learn more about RP through an introductory page and a machine list can be brought up. This list contains all the machines in the current database and provides links to the websites for the manufacturers (Fig. 4.4b).

Consider a mobile phone manufacturer who wishes to use RP to help in designing concept modelling. In addition the manufacturer might have a problem in visualizing and costing for moulding patterns from outside toolmakers. The approximate size of the models is known (30 mm × 15 mm × 130 mm) and the machine is to be installed in an office environment in a space no larger than 1.2 m × 1.5 m. The manufacturer has engineering support and understanding but no-one with a specific background in RP. The problem is where to start looking to find out a little more on the subject of RP and the available technology before contacting vendors to evaluate the technology more closely.

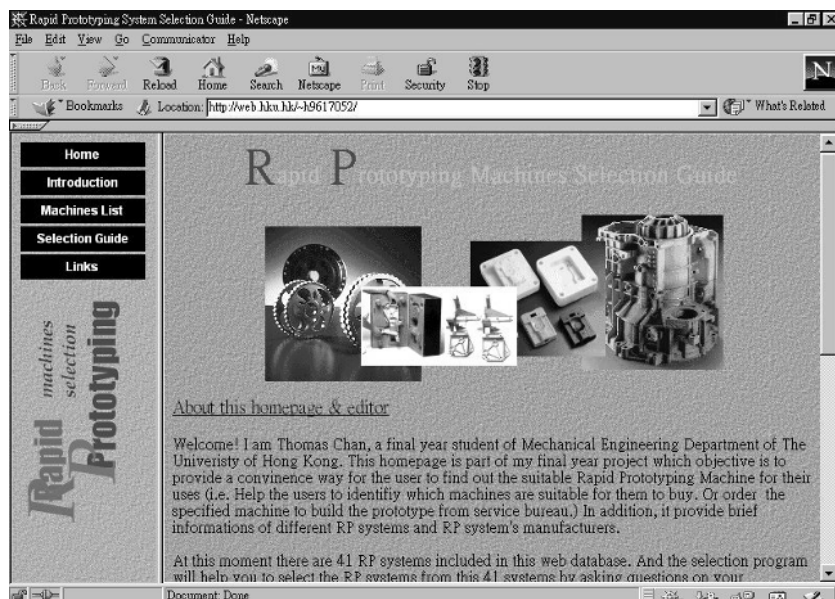


Fig. 4.4a The opening page of the DSS

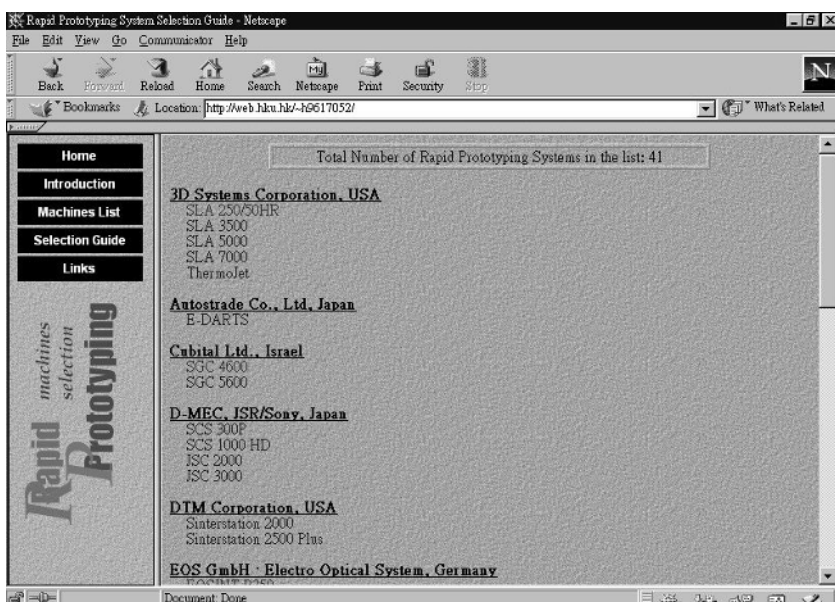


Fig. 4.4b The machine list



The user is then led through the DSS part of the program. This is constructed in the form of a series of sheets associated with each rule area. Figure 4.4c shows the sheet associated with the characteristics of the prototype. The user can input values for selected parameters, some of which are numerical and some of which indicate level of importance. The question on materials really relates to which materials would be acceptable to the user. In the case marked there is no machine that produces both wax and paper products. The answer would give machines that use either wax or paper. Currently, there is no cross-referencing between questions and so the issue relating to heat resistance is not queried in terms of material properties. For instance, wax has a relatively low heat resistance and strength compared with other materials and yet it is possible to label wax as an acceptable material, while maintaining strength and heat resistance as being important criteria. The boxes marked with question marks (?) provide the user with additional information relevant to that question. In here, guidance as to what would be meant by medium heat resistance (for example) would be given to the user. Since the criteria are somewhat subjective, it is clear that it is possible to fool this system. However, this system complies with the garbage-in garbage-out rule and proper use will hopefully ensure the user is better informed after a number of runs.

Dividing the DSS into sheets means that it is possible to adjust the weighting system in terms of importance. If it does not matter what material is being used or what the installation requirements are, but it does matter that the parts are used for specific applications, then it may be appropriate to adjust the application weighting higher than the other sheets (Fig. 4.4d). These are calculated relative to each other and so the absolute setting is not critical.

Once the user has input data, it is possible to perform a search (Fig. 4.4e). Based on the mobile phone manufacturer scenario, the results are for FDM machines, followed by Thermojet, followed by the Modelmaker machine. Note that this system was originally built around an incomplete database. For example the ZCorp machine was not included. However, based on the input criteria this selection seems reasonable. The user is not encouraged, however, to explicitly interpret a single result or absolute value. Rather it is better to run the program a number of times to see what decision affects the selection.

**Rapid Prototyping Machines Selection** [Prev] [Next] [Search]

Prototype Application Working Environment Preference

1. The envelope size of the largest part of the prototype (Unit mm):  
 X 30 Y 15 Z 30

2. Which of the following materials will you choose for prototyping?  
☒ Paper ☐ Resin ☐ Polyamide ☐ Nylon  
☐ ABS ☒ Wax ☐ Polystyrene ☐ Metals  
☐ Ceramics ☐ Sand ☐ Polycarbonate ☐ Other

3. Resistance of the prototype:  
 Heat Resistance: ☐ Low ☒ Medium ☐ High  
 Water Resistance: ☒ Low ☐ Medium ☐ High  
 Chemical Resistance: ☒ Low ☐ Medium ☐ High

4. Resolution required: ☐ Low ☒ Medium ☐ High

5. Surface Finish Quality: ☐ Low ☒ Medium ☐ High

6. Strength of the prototype: ☐ Weak ☐ Fair ☒ Strong

Fig. 4.4c Prototype characteristics sheet

**Rapid Prototyping Machines Selection** [Prev] [Next] [Search]

Prototype Application Working Environment Preference

In order to optimize the selection process for you, you can adjust the weighting among different areas which contribute to the RP machines selection.

**Level of Importance**

Less important Most important

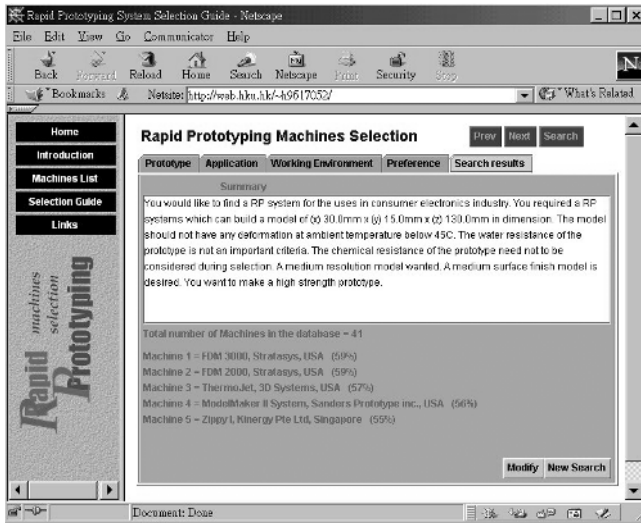
Prototype: 1 2 3 4 5 6 7 8 9 10 (Slider at 6)

Application: 1 2 3 4 5 6 7 8 9 10 (Slider at 8)

Working Environment: 1 2 3 4 5 6 7 8 9 10 (Slider at 4)

NOTE:  
 In general, 'Application' is the dominant factor among all, so it was set to highest rate in default. If you don't have any special condition/limitation, you are recommended to use the default setting.

Fig. 4.4d Adjusting weightings



**Fig. 4.4e Results page**

#### 4.3.4 Connection to database

It has been stated that this web-based tool would only work as well as the data held within the database and that this should contain all commonly used RP systems. The information for each machine should be in a standard form and all questions should relate in some way to each machine. The system was originally developed with a stand-alone database but an early objective was to integrate it with the database mentioned in Chapter 2 of this book. This database is designed to hold information on all RP machines and be updateable from the web, making it easy to incorporate new machine data as soon as it is available. The database is available in Microsoft Access format and has been tested using Java 2 SDK for accessing the information.

All of the system questions are available from the database, except those relating to chemical, heat, and water resistance. These can be included within a section of the materials portion of the database.

#### 4.3.5 Further modification of the system

There are two forms of experience associated with this system. Experience leads to advice and it is good advice that the user is hoping to obtain in order to avoid making a costly mistake. The first form of experience relates to individual use of a machine. This is designed to

provide the user with insight as to how to use the machine. Knowing that some machines use support structures, for example, indicates that time and resources must be allocated for generation of supports during build set-up and removal of the supports afterwards. Similarly, a wide range of available materials indicates that a machine may be suitable for many applications but then it may be costly to keep a stock of all those materials and it may be troublesome to switch from one material to another. This kind of experience can be stored in relation to a single machine or process within the database. The difficulty arises from determining the experience and generating the advice. For this kind of system, it is probably more important than the numerical data. Placing the advice in relation to different aspects of the system is not a problem. Providing consistent and unbiased advice is however, and the system requires much further modification to increase the amount of advice and to verify this advice, both against potential users and acknowledged experts in the field.

The second form of experience relates to comparative experience. In this case it may be that a certain machine is faster or a certain material is better. This kind of information is dynamic in nature and will change as new machines and materials are introduced. It is not within the scope of this system to provide such comparisons in the form of advice. The system cannot say 'You must buy machine A because it is faster than machine B and you have said you want the fastest machine.' Such comments are too subjective and it must be up to the user to make those comparisons.

#### **4.4 PRODUCTION PLANNING AND CONTROL**

This is an extension of the first problem and is probably most relevant to the activities of service bureaus (SBs) or modelmakers that operate one or more RP machines and processes. The SB may know what machine and material a part is to be made from, but in most circumstances the part cannot be considered in isolation. When any new part is presented to the process planner at the SB, it is likely that he has already committed to build a number of parts. A decision support software system may be useful in keeping track and optimizing machine utilization.

Consider the process when a new part is presented to the SB for building. In general, the information presented to the process planner

will include the following:

- Part geometry
- Delivery date
- Processes other than RP to be carried out (pre-processing and post-processing)
- Expectations of the user (accuracy, degree of finish, etc.)

Furnished with this small amount of data, it is possible to start integrating the new job with the existing jobs and available resources.

#### **4.4.1 Pre-processing**

Pre-processing means software-based manipulation. This will be carried out on the file that describes the geometry of the part. Such manipulation can generally be divided into two areas: modification of the design and determination of build parameters.

Modification of the design will be carried out on the model data as provided by the customer. It is generally assumed that the model initially comes from an outside source. For a large organization, this may be a different department. However it is seen, the RP user is providing a service to an outside source. Such design modification may be further divided into two areas:

- Addition of design detail. This kind of work is often carried out by SBs in which the design is modified to suit the user requirements. This may range from the addition of a small design feature to modification of the design to suit a manufacturing process (like creating tooling from a master pattern).
- Repair of design problems. These are more specific to the destined RP process and are particularly relevant to occasions where the design may not meet the user expectations in some way. Examples here include repair of STL files, smoothing of heavily faceted data sets, etc.

Determination of the build parameters is very specific to the RP process to be used. This includes support generation, setting of build styles, layer thickness selection, temperature setting, etc. In general this is either a very quick process or it takes a predictable length of time to set up. On occasion, and for some particular types of machine, this process can be very time consuming. This usually corresponds to instances where the user expectations closely meet the upper limits of the machine specification (high accuracy, build strength, early delivery

date, etc.). Under such conditions the user must devote more time and attention to parameter setting. The decision support software should make the process planner aware under which circumstances this may occur and allocate resources appropriately.

#### **4.4.2 Part build**

For some processes, like FDM, it does not really matter in terms of time whether you build one part after another or group parts together in batches. Most processes, however, will vary significantly regarding this factor. This may be due to significant preparation time before the build process takes place (like waiting for temperatures in SLS), or because there is a significant delay between layers. In the latter case it is obvious that the cumulative number of layers should be as low as possible to minimize the overall build time for many parts, thus optimizing the machine.

Another factor is part orientation. It is well known that because of heterogeneous behaviour of all RP processes according to the  $x$ -,  $y$ -, and  $z$ -directions, parts will generally build more effectively in one orientation compared with another. This can cause difficulties when organizing the batch production of parts. Orientation of parts so that they fit efficiently within the work volume does not necessarily mean optimal build quality and vice versa. For those machines that need to use support structures during the build process, this represents an additional problem, both in terms of build time (allocation of time to build the support structures for different orientations) and post-processing time (removing the supports). Many researchers have discussed these dilemmas (19–23).

What this generally means to a process planner is compromise. Compromise is not unusual to a process planner – in fact it is a typical characteristic – but the degree of flexibility provided by many RP machines makes this a particularly interesting problem. It can be seen from the above discussion that the RP process can appear to be a bottleneck even when perhaps it is not. Just because an RP machine is being used constantly does not mean it is being used efficiently.

#### **4.4.3 Post-processing**

All RP parts require a degree of post-processing. At the low end this may require removal of support structures or excess powder for those

who merely want quick, simple verification. At the high end, the RP process may be a very insignificant time overhead in the overall process. Parts may require a large amount of skilled manual work in terms of surface preparation and coating. Alternatively, the RP part may be one stage in a complex rapid tooling process that requires numerous manual and automated stages. All this can result from the same source machine. It can even be an iterative process involving all of the above steps at different stages in the development cycle based on the same part CAD data.

#### **4.4.4 Summary**

It is clear that only process planners who have a very detailed understanding of all the roles that RP parts can play will be able to utilize the resources effectively and efficiently. Even then it may be difficult to perform this task reliably given the huge number of variables involved. A software system to assist in this difficult task would be a very valuable tool.

Decision support software for processing planning must incorporate a wide range of expertise and understanding of the actual process. Setting up, verifying, and updating such a system must be easy and straightforward in order to make it possible to tailor it to individual needs.

### **4.5 OPEN PROBLEMS**

Some open problems that motivate continued research are presented here.

- Selection systems are only as good as the information that is utilized to make suggestions. Maintaining up-to-date and accurate machine and material databases will probably be an ongoing problem. Centralized databases and standard database and benchmarking practices will help to mitigate this issue.
- Customers (people wanting parts made) have a wide range of intended applications and needs. A better representation of those needs is required in order to facilitate better selection decisions. Improved methods for capturing and modelling user preferences are also needed.
- Related to the wide range of applications is the wide range of manufacturing process chains that could be used to construct prototypes. Better and more complete methods of generating,

evaluating, and selecting process chains are needed for cases where multiple parts or products are needed (10 to 100) or when complex prototypes need to be constructed. An example of the latter case is a functional prototype of a new product that consists of electronic and mechanical subsystems. Many options probably exist for fabricating individual parts or modules.

- More generally, integration of selection methods, with databases and process chain exploration methods, would be very beneficial.
- Methods are needed that hide the complexity associated with the wide variety of process variables and nuances of RP technologies. This is particularly important for novice users of RP machines or even for knowledgeable users who work in production environments. Alternatively, knowledgeable users must have access to all process variables if necessary to deal with difficult builds.
- Better methods are needed that enable users to explore trade-offs (compromises) among build goals and to find machine settings that enable them to best meet their goals. These methods should work across the many different types of RP machines and materials.
- It is not uncommon for RP customers to want parts that are at the boundaries of RP machine capabilities. Tools that recognize when capability limits are reached or exceeded would be very helpful. Furthermore, these tools should provide guidance that assists users in identifying process settings that are likely to yield the best results. Providing estimates of part qualities (e.g. part detail actual sizes versus desired sizes, actual surface finish versus desired surface finish, etc.) would also be helpful.

## ACKNOWLEDGEMENTS

Dr Rosen acknowledges the US National Science Foundation for grant DMI-9618039, and the RPMI member companies at Georgia Tech for their support. In addition, Marco Fernandez, a PhD student at Georgia Tech, deserves thanks for assisting with the example in Section 2.

## REFERENCES

- 1 Wohlers, T. (2000) Rapid Prototyping: State of the Industry – 1999 Worldwide Progress Report. Wohlers Associates, Inc., 2000.



- 2 Deglin, A. and Bernard, A. (2000) A knowledge-based environment for modelling and computer-aided process planning of rapid manufacturing processes. In CE 2000 Conference, Lyon, France, July.
- 3 Xu, F., Wong, Y. S., and Loh, H. T. (1999) A knowledge-based decision support system for RP&M process selection. In Proceedings of the Solid Freeform Fabrication Symposium, Austin, Texas, 9–11 August.
- 4 Mistree, F., Smith, W. F., Bras, B. A., Allen, J. K., and Muster, D. (1990) Decision-based design: a contemporary paradigm for ship design. *Transactions of the Society of Naval Architects and Marine Engineers*, **98**, 565–597.
- 5 Mistree, F., Hughes, O. F., and Bras, B. A. (1993) The compromise decision support problem and the adaptive linear programming algorithm. In *Structural Optimization: Status and Promise* (Ed. M. P. Kamat), pp. 247–286 (AIAA, Washington, DC).
- 6 Mistree, F., Smith, W. F., and Bras, B. A. (1993) A decision-based approach to concurrent engineering. In *Handbook of Concurrent Engineering* (Eds H. R. Paresai and W. Sullivan), pp. 127–158 (Chapman & Hall, New York).
- 7 Marston, M. C. (2000) Game based design and multi-disciplinary optimization. PhD dissertation, School of Mechanical Engineering, Georgia Institute of Technology.
- 8 Marston, M., Allen, J. K., and Mistree, F. (2000) The decision support problem technique: integrating descriptive and normative approaches. *Engineering Valuation and Cost Analysis, Special Issue on Decisions-Based Design: Status and Promise*, **3**, 107–129.
- 9 Simon, H. A. (1996) *The Sciences of the Artificial* (MIT Press, Cambridge, Massachusetts).
- 10 Bascaran, E., Bannerot, R. B., and Mistree, F. (1989) Hierarchical selection decision support problems in conceptual design. *Engineering Optimization*, **14**, 207–238.
- 11 Herrmann, A. and Allen, J. K. (1999) Selection of rapid tooling materials and processes in a distributed design environment. In ASME Design For Manufacturing Conference, Las Vegas, 12–15 September, paper No. DETC99/DFM-8930.
- 12 Keeney, R.L. and Raiffa, H. (1976) *Decisions with Multiple Objectives: Preferences and Value Tradeoffs* (John Wiley, New York).

- 13 von Neumann, J. and Morgenstern, O. (1947) *The Theory of Games and Economic Behavior* (Princeton University Press, Princeton, New Jersey).
- 14 Luce, R. D. and Raiffa, H. (1957) *Games and Decisions* (John Wiley, New York).
- 15 Savage, L. J. (1954) *The Foundations of Statistics* (John Wiley, New York).
- 16 Thurston, D. L. (1991) A formal method for subjective design evaluation with multiple attributes. *Research in Engineering Design*, **3**(2), 105–122.
- 17 Hazelrigg, G. (1996) *Systems Engineering: An Approach to Information-Based Design* (Prentice Hall, New Jersey).
- 18 Fernandez, M. G., Seepersad, C. C., Rosen, D. W., Allen, J. K., and Mistree, F. (2001) Utility-based decision support for selection in engineering design. ASME Design Automation Conference, Pittsburgh, 9–12 September, paper DETC2001/DAC-21106.
- 19 Allen, S. and Dutta, D. (1995) Determination and evaluation of support structures in layered manufacturing. *Journal of Design and Manufacturing*, **5**, 153–162.
- 20 Cheng, W., Fuh, J. Y. H., Nee, A. Y. C., Wong, Y. S., Loh, H. T., and Miyazawa, T. (1995) Multi-objective optimization of part building orientation in stereolithography. *Rapid Prototyping Journal*, **1**(4), 12–23.
- 21 Lan, P.-T., Chou, S.-Y., Chen, L.-L., and Gemmill, D. (1997) Determining fabrication orientations for rapid prototyping with stereolithography apparatus. *Computer-Aided Design*, **29**(1), 53–62.
- 22 Thompson, D. C. and Crawford, R. H. (1995) Optimizing part quality with orientation. Solid Freeform Fabrication Symposium (Eds H. L. Marcus *et al.*), University of Texas, Austin, August.
- 23 West, A. P., Sambu, S., and Rosen, D. W. (2001) A process planning method for improving build performance in stereolithography. *Computer-Aided Design*, **33**(1), 65–80.

*This page intentionally left blank*

# Chapter 5

## Adaptive slicing for rapid prototyping

**R-S Lin**

Department of Mechanical Engineering, National Chung Cheng University, Taiwan, Republic of China

### ABSTRACT

Rapid prototyping (RP) has been widely used in industry for many years. Using rapid prototyping technologies to improve the manufacturing lead-time has been demonstrated widely in the literature. The slicing operation is one of the major processes in the CAD/CAM/RP system, which transforms a three-dimensional part model into two-dimensional layers. RP fabricates the part model layer by layer; this has the advantage of simplicity for two-dimensional manufacturing instead of the complexity of three-dimensional manufacturing. In recent years, some of the RP research has focused on improving the fabrication accuracy and efficiency. In general, the part accuracy depends upon the number of layers being used. Although using large numbers of thin layers can improve surface finish, it makes the RP fabrication inefficient. The adaptive slicing technique determines the minimum number of layers required under an allowable stair-stepping error for the RP part; therefore, it reduces the fabrication time. The criterion for determining layer thickness is based on the relationship between the part geometry and the user-defined surface accuracy. This chapter discusses the fundamental adaptive slicing algorithms as well as the newly developed advanced adaptive slicing. The advanced approach is an improvement based on the fast interior

and accurate exterior method. To implement the adaptive slicing techniques, the RP machine must have the capability of changing the layer thickness during material deposition. The analyses for RP fabrication time show that the advanced adaptive slicing method is seven times faster than conventional constant layer slicing under the same surface accuracy requirement for a demonstrated example.

## 5.1 INTRODUCTION

Rapid prototyping (RP) technology, a non-conventional manufacturing process, has been widely used in industry recently for prototypes, mechanism interference checking, conceptual design, and rapid tooling. Currently there are many RP systems available in industry, such as fused deposition modelling (FDM), selective laser sintering (SLS), stereolithography apparatus (SLA), three-dimensional printing, and Model Maker. The basic difference between these technologies is the methods by which raw materials are prepared. The commonly used commercially available RP materials are wax, PVC, ABS, and metal powders. Depending upon the material type, some technologies require lasers for heat energizing. Some use adhesives for binding. Some RP systems need to build support for overhanging part fabrication. Although these technologies differ, the underlying principle for the CAD/CAM data processing for most RP systems is the same. The common process involves slicing the three-dimensional part model into two-dimensional layers.

The slicing process cuts the three-dimensional model into layers and obtains a series of two-dimensional contours. The simplest method is the constant layer slicing. This method uses a constant-thickness layer approach, but the remaining surface stair-stepping errors are varied according to the surface geometry changes. Dolenc and Mäkelä (1) presented the basic adaptive slicing algorithm in which the staircase errors were under a user-specified tolerance. Kulkarni and Dutta (2) developed an adaptive slicing algorithm for parametrizable algebraic surfaces that resulted in varying layer thickness and satisfied the specified cusp height error. Sabourin *et al.* (3) presented the stepwise uniform refinement method, which is an improvement of the basic adaptive slicing method. In 1997, the same group proposed the accurate exterior and fast interior method (4). The thick layers were chosen for material deposition on the interior, and thin layers for the exterior. Tata *et al.* (5) proposed an efficient slicing algorithm to increase RP

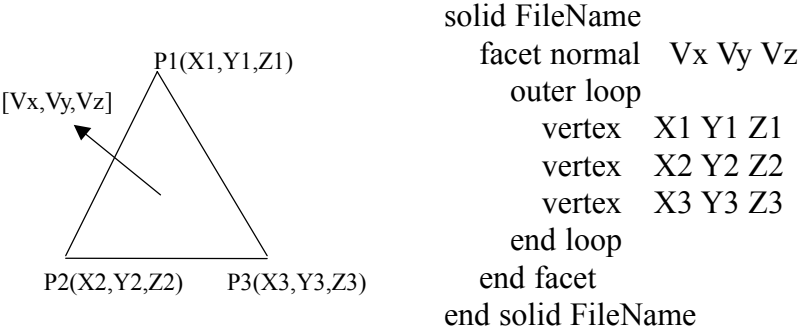
productivity. Tyberg and Bohn (6) presented the local adaptive slicing method, which dealt with the part having different features. To improve the surface finish, Hope *et al.* (7) used the five-axis machining techniques to remove the RP staircases. Some researchers (8–12) focused on the optimal orientation for RP fabrications, which determined the best build orientation with fewer layers.

After the sliced contour for each layer is obtained, the RP machine deposits material according to the two-dimensional contours on each layer. Normally, the material deposition paths consist of the offset paths and the zigzag paths. Some tool path planning methodologies developed for CNC two-dimensional machining can be applied to RP fabrications. Farouki *et al.* (13) discussed the offset curve algorithm for piecewise linear/circular boundary contours.

## 5.2 CAD/CAM/RP DATA PROCESSING

This chapter discusses the development of adaptive slicing algorithms and the implementation on CAD/CAM/RP systems. The slicing operation is one of the major processes for CAD/CAM/RP systems, which decomposes a three-dimensional model into sequential two-dimensional layers. RP processing normally begins with the input of a three-dimensional model in STL format, which becomes a standard format for most RP systems. The STL format is a tessellation representation, which defines a three-dimensional object by a series of triangular facets. Each triangular facet is defined by three vertices and a unit normal vector for the facet where the normal vector direction indicates the outside of the object. Figure 5.1 shows the data format of a triangular facet. The Cartesian co-ordinates system for three vertices are  $(X1, Y1, Z1)$ ,  $(X2, Y2, Z2)$ , and  $(X3, Y3, Z3)$ . The facet normal vector is  $(V_x, V_y, V_z)$ .

In order to simplify the slicing processes and reduce the computational load, topological information is added into the data structure. The topological structure from top to bottom includes polygons (for triangular facets and contours), edges, and vertices. The basic member for this structure is the geometric information of vertices, or ‘called point’. Using this data structure, the geometric information (e.g. vertex co-ordinates) is recorded only once. If the program needs any geometric information, the pointer embedded within the data structure will indicate the desired address for the particular information.



**Fig. 5.1 The triangular facet and its data format**

In general, RP fabrications build the part layer by layer from the bottom layer to the top. This process requires the two-dimensional contour geometry of each layer. The contours can be obtained by slicing the three-dimensional part along the varied *z*-planes; this is called the slicing process. The slicing process includes the following three steps: (a) calculating the intersections between the object and a horizontal plane, (b) sorting intersection points, and (c) constructing two-dimensional contours. Some RP systems require building supports before depositing the next layer if the STL model contains overhanging parts. The following sections describe these algorithms in detail.

**5.2.1 Calculating the intersection points**

The linear interpolation algorithm is used to calculate the intersection points between the model and the sliced *z*-plane, where the height of the *z*-plane is *z*<sub>sliced</sub> (*z* = *z*<sub>sliced</sub>). The mathematical expression for this algorithm is shown in equation (5.1). The sliced *z*-plane intersects with an edge, whose vertices are *P*1(*x*1,*y*1,*z*1) and *P*2(*x*2,*y*2,*z*2), of a triangular facet.

$$x = x_1 + (z_1 - z_{\text{sliced}})/(z_1 - z_2) \times (x_2 - x_1)$$
$$y = y_1 + (z_1 - z_{\text{sliced}})/(z_1 - z_2) \times (y_2 - y_1) \quad \text{where } z_1 \leq z_{\text{sliced}} \leq z_2$$
$$z = z_{\text{sliced}}$$

(5.1)

### 5.2.2 Sorting the intersection points

The above intersection points need to be sorted before the layered two-dimensional contours can be constructed, since these points are not in sequence. The main idea for the sorting algorithm uses the embedded topological information. In the previously mentioned topological data structure, each triangular facet (or polygon) and edge have been assigned an integer number, as a pointer. Therefore, the calculated intersection data include not only the geometrical information, but also the polygon number and edge number to which the intersection point belongs. Since the intersection data include the polygon number information, the next intersection point for the sorting process is determined by checking the same polygon number among the intersection points. The next step uses the co-edge information, since each triangular facet in a closed object shares an edge with another facet. Therefore, the next intersection point can be chosen by checking the same edge number. This sorting process will be continued until the end of the intersection points.

### 5.2.3 Constructing a two-dimensional contour

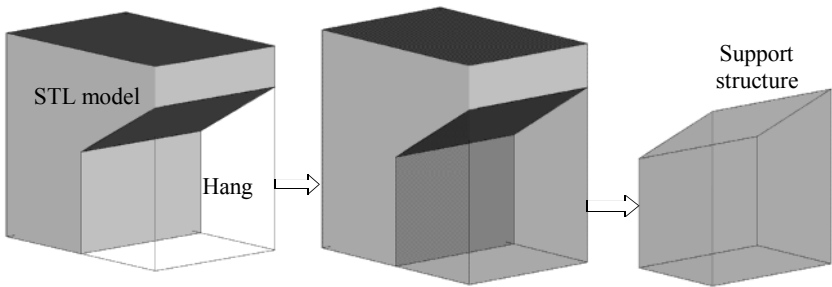
The final step in the slicing process is to record these intersection points as a contour loop, called two-dimensional contour construction. The loop-direction information, either clockwise (CW) or counter clockwise (CCW), is important for the next process, material deposition path planning. Conventionally, the CW and CCW directions indicate the inner and outer loops, respectively. The correct contour loop direction can be determined by checking the vector (cross) product of the triangular facet normal vector and the vector formed by any two intersection points of the same facet. If the  $z$ -component of the above vector product is greater than zero, the vector direction formed by these two intersection points is correct for conventional notation. If less than zero, the contour loop direction needs to be reversed for contour data recording.

To check if a contour loop is inner or outer, the odd winding rule method is used. This method draws a straight line from any point on the tested contour to the right. If this line intersects all the contours in a layer with odd number, the tested contour is an outer loop. Otherwise, it is an inner loop. During the two-dimensional contour construction, some intersection points may be located in the same straight line, called co-linear. The redundant points need to be

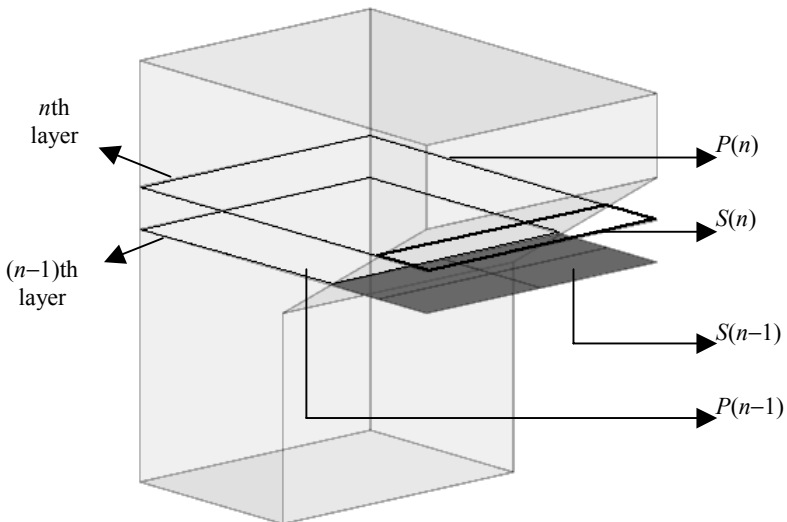




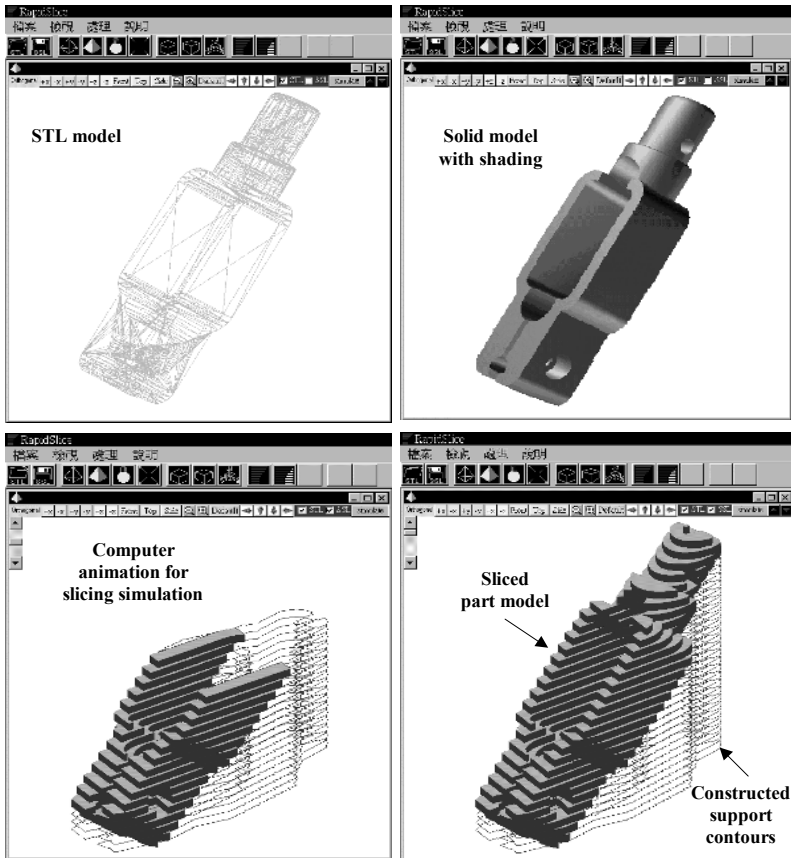
where the operator ‘+’ is the union operation and ‘-’ is the difference.  $S$  and  $P$  represent the support contour and the model contour, respectively. Figure 5.4 illustrates the support contour in the dark area for the  $(n-1)$ th layer,  $S(n-1)$ . If the model has been sliced for  $m$  layers and the  $m$ th layer is on top, the support area  $S(m)$  is 0. Using the recursive method for equation (5.2), the support contours for each layer can be determined where  $n$  is from  $m$  to 1. Figure 5.5 shows the sliced results and the RP fabrication simulation processes of an STL



**Fig. 5.3 An overhanging model and the support structure**



**Fig. 5.4 Illustration of the support construction by Boolean operation**



**Fig. 5.5 The computer simulation program for RP fabrication processes of a demonstrated example**

part model. This RP fabrication simulation is a computer graphics animation program, which generates deposited material layer by layer. In order to see the generated support layers clearly, the part model is fabricated with an inclined angle, and is sliced with very thick layers. The contours shown in the bottom panels are support contours and the grey shaded layers are the simulation for RP model depositions.

### 5.3 FUNDAMENTALS OF ADAPTIVE SLICING

The slicing process for RP fabrications produces a part with cusps, or stair-stepping error, remaining on the part surface. Normally the cusp height can be reduced if the model is sliced with thinner layers;

however, this causes longer RP fabrication time. The adaptive slicing approach improves both the forming accuracy and the fabrication efficiency, compared with the conventional constant thickness layer slicing method. The forming accuracy depends on the stair-stepping errors, due to the slicing process. The fabrication efficiency depends on the RP fabrication time, which assumes a constant speed tool motion for the material deposition. This chapter discusses two adaptive slicing algorithms: (a) basic adaptive slicing and (b) advanced adaptive slicing.

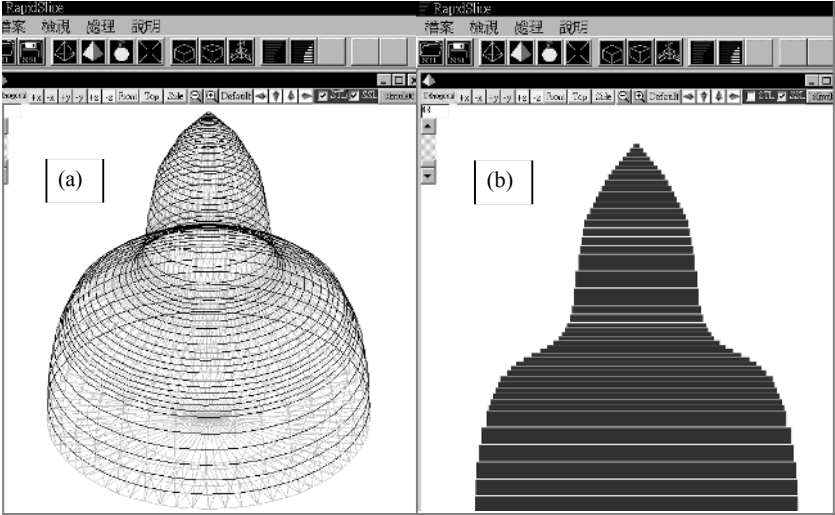
### 5.3.1 Basic adaptive slicing

In order to keep a constant stair-stepping error on the RP part surface, adaptive slicing considers the part surface geometry when determining the slicing layer thickness. The outputs for adaptive slicing are two-dimensional contours with various layer thicknesses. The variations in layer thickness according to the changes in part surface geometry keep the surface stair-stepping errors constant. In contrast, the constant layer slicing method results in varied stair-stepping errors on the part surface.

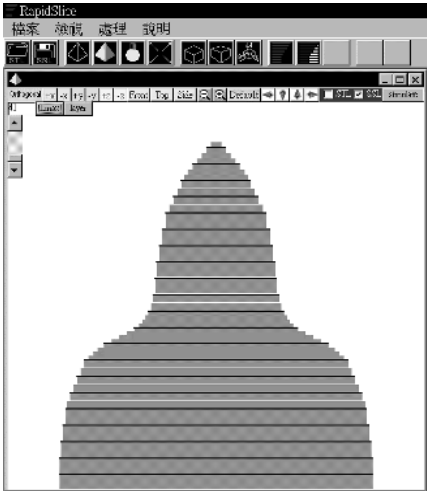
#### 5.3.1.1 Determination of layer thickness

The layer thickness determination for adaptive slicing depends on the surface geometry for the part (STL) model and the RP machine capability for changing the deposition layer thickness. The assumption for this study is that the RP machines can change their deposition thickness between  $L_{\min}$  and  $L_{\max}$ , which are the constraints for determining the suitable layer thickness. Under the user-specified allowable cusp, the layer thickness is equal to the cusp height divided by the  $z$ -component of the unit normal vector of the triangular facet. To satisfy the surface accuracy requirement, or stair-stepping error requirement, the smallest layer thickness is usually chosen for the entire layer. The above calculated layer thickness must be greater than  $L_{\min}$ , the thinnest layer that the RP machine can provide. If not, the RP machine uses  $L_{\min}$  as the determined layer thickness.

A demonstrated part model sliced by the basic adaptive slicing algorithm is shown in Fig. 5.6. Figure 5.6a shows the three-dimensional view of the STL model in grey and the sliced results of the two-dimensional contours in black. Figure 5.6b shows the layer deposition results in two-dimensional view. The result shows that the determined layer depends on the model's surface geometry.



**Fig. 5.6 (a) The adaptive slicing results and the SLT model;  
(b) the layer deposition in two-dimensional view**



**Fig. 5.7 The sliced results using the improved adaptive  
slicing method**

An improved method, called stepwise uniform refinement (3), was modified from the above basic adaptive approach. The improved method can detect fine features on the part model. This algorithm first sliced the model with the RP machine allowable maximum layer thickness ( $L_{\max}$ ). Then, each thick layer was sliced again with thinner layers. The final layer thickness was determined by checking both the top and bottom surface normals of a layer. The sliced result achieved using the improved method for the same example is shown in Fig. 5.7. The horizontal black lines shown in Fig. 5.7 represent the thick layer separations. Each thick layer was further separated into thinner layers according to the surface geometry changes.

### 5.3.2 Analyses for basic adaptive slicing

Adaptive slicing controls the surface accuracy, or stair-stepping errors, by varying the layer thickness according to the surface geometry changes. To examine the RP deposition accuracy, the percentage volumetric error,  $E_v$ , is defined by

$$E_v = [(V_p - V_m)/V_m] \times 100\% \quad (5.3)$$

where  $V_m$  is the total volume of the STL model and  $V_p$  is the volume of the deposited RP part. The volume  $V$  can be calculated by

$$V = \sum_{i=1}^N A_i H_i \quad (5.4)$$

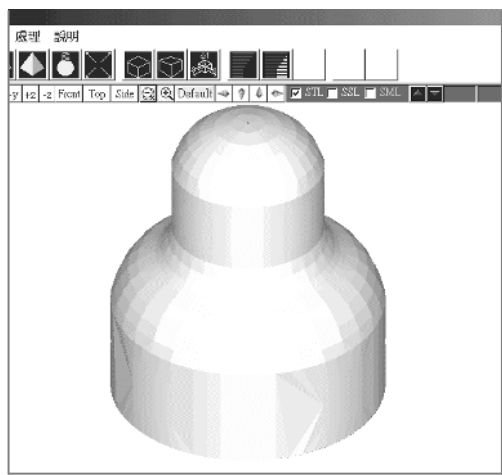
where

$$A_i = \frac{1}{2} \sum_{j=1}^{m+1} (x_j y_{j+1} - x_{j+1} y_j) \quad (5.5)$$

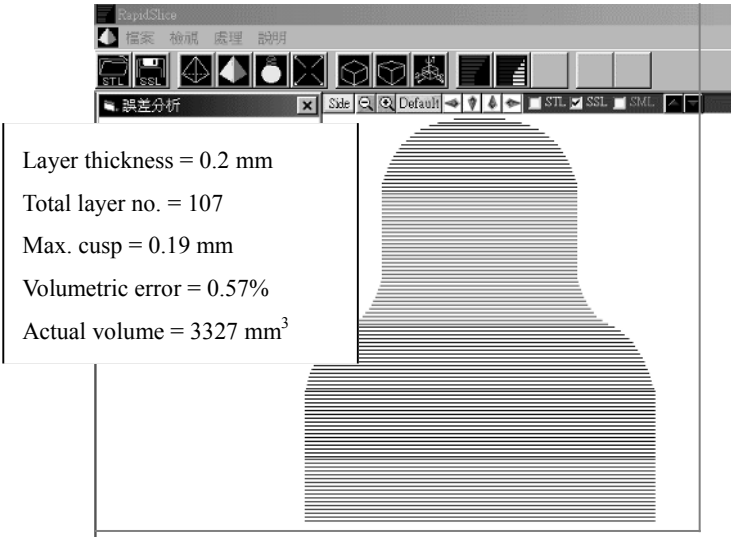
where  $H_i$  is the thickness for the  $i$ th layer,  $A_i$  is polygon (contour) area for the  $i$ th layer,  $m$  is the polygon edge number of a contour, and  $x$  and  $y$  are the polygon vertices co-ordinates. If  $A_i > 0$ , the (contour) polygon is an outer loop; otherwise, it is an inner loop.

A demonstrated example shown in Fig. 5.8 is used for comparison with two kinds of slicing algorithms, constant layer slicing and basic adaptive slicing. The assumption for this adaptive slicing process is that the deposition layer thickness can be chosen from 0.15, 0.2, 0.3, and

0.6 mm. Figure 5.9 shows the sliced results achieved using constant layer processing. To keep the maximum cusps under 0.19 mm and the percentage volumetric error at 0.57 per cent, 0.2 mm layer thickness is used. The constant layer slicing method requires 107 layers in total to finish this fabrication process. Using the same example, the adaptive



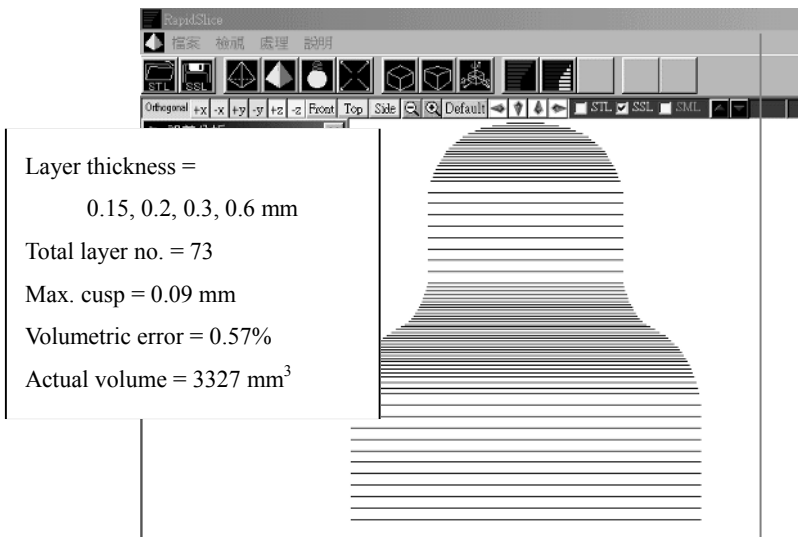
**Fig. 5.8 The demonstrated example for different slicing processes**



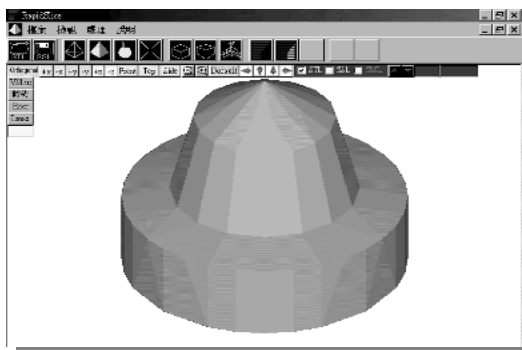
**Fig. 5.9 Slicing results using the constant thickness layer slicing method**

slicing method takes only 73 layers to keep the same percentage volumetric error of 0.57 per cent. Therefore, the adaptive slicing process saves 34 layers of fabrication time. The maximum cusps remaining on the part surface are 0.09 mm using adaptive slicing. Figure 5.10 shows the results of adaptive slicing and error analyses.

Another part example, shown in Fig. 5.11, is used to demonstrate the slicing result differences between the basic adaptive slicing and the



**Fig. 5.10 Slicing results using the basic adaptive slicing method**



**Fig. 5.11 The part used for comparison of the basic and improved adaptive slicings**

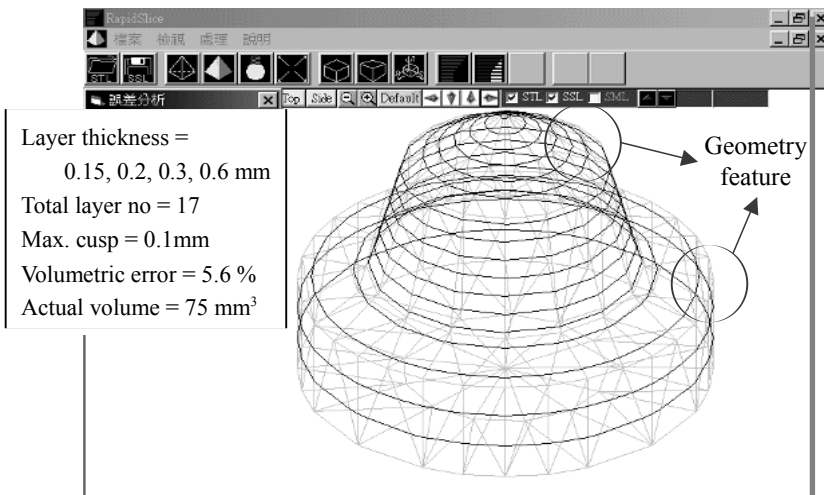


improved adaptive slicing methods. To keep to the user-specified allowable cusp height of 0.1 mm, it requires 17 layers for RP fabrication using the basic adaptive slicing method. The sliced result is shown in Fig. 5.12.

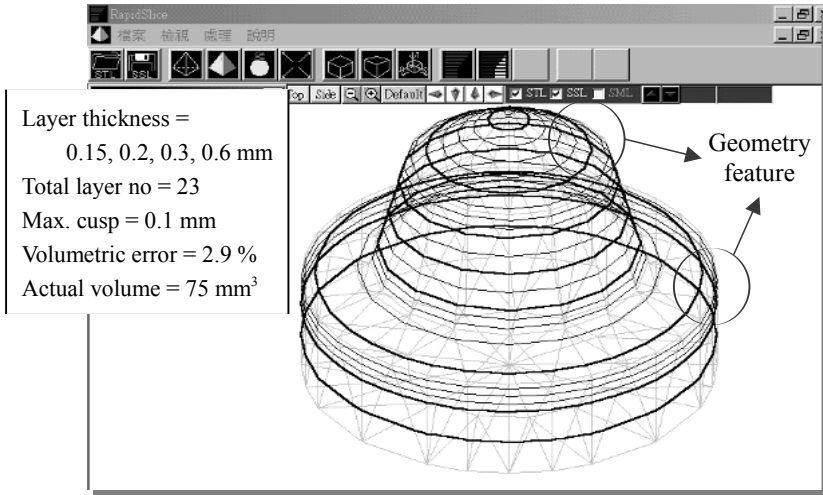
Under the same surface accuracy conditions, Fig. 5.13 shows the improved adaptive slicing results. The layer number required for this method is 23, which is larger than the basic approach. The reason for the increasing layer number is that this part has a rapidly changing surface slope in some areas, as marked by circles in Figs. 5.12 and 5.13. The improved approach detects this geometry feature and slices out with more layers. The improvement results in a percentage volumetric error of 2.9 per cent, compared with 5.6 per cent for the basic adaptive slicing approach.

### 5.3.3 Problems for current adaptive slicing

Many adaptive slicing algorithms have been developed to improve fabrication efficiency under a specified surface accuracy, or allowable stair-stepping error. The result is highly dependent on the part surface geometry (surface slopes or normals). The flatter the surface normal, the thicker the layer will be. The minimum layer thickness was usually chosen for the entire layer fabrication, if the surface slopes were not constant in one layer. The reason for choosing the minimum thickness



**Fig. 5.12 Slicing results from the basic adaptive slicing method**



**Fig. 5.13 Slicing results from the improved adaptive slicing method**

is that it satisfies the slicing accuracy requirement easily. However, this makes the RP fabrication process inefficient. Figure 5.14 shows a part model with this situation. To satisfy the stair-stepping error requirement, the layer thickness determination is based on the surface slope located in the C1 area, which has a smaller surface slope. However, this solution makes the RP fabrication process inefficient in the C2 area.

## 5.4 ADVANCED ADAPTIVE SLICING

To make the RP fabrication process efficient, the fast interior and accurate exterior method (4) is utilized for the initial process of the advanced adaptive slicing method. A demonstrated part processed using the fast interior and accurate exterior method is shown in Fig. 5.15. This method separates the model into the interior and exterior areas. The interior area uses thick layers to speed up the fabrication process. The outside boundary uses thin layers to obtain better surface finish. This section will discuss an advanced adaptive slicing method (14), which improves the fast interior and accurate exterior method. The main idea behind this approach is that the separated outside boundary can be further separated into different segments according to the geometry changes.

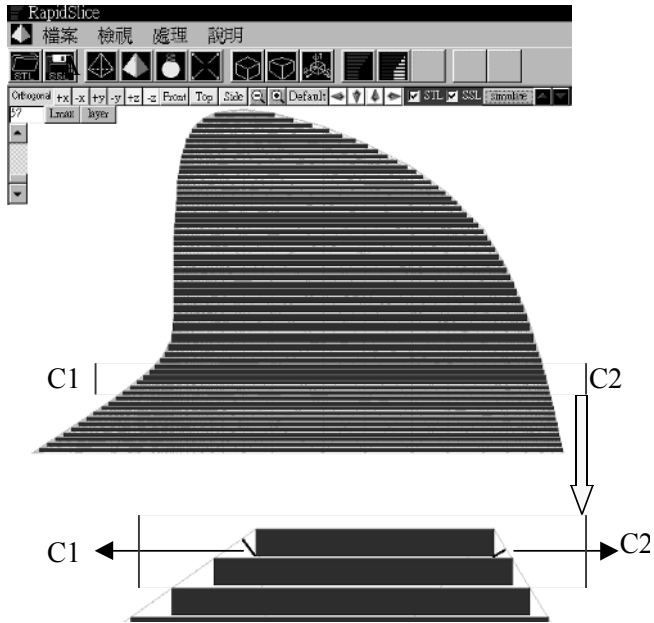


Fig. 5.14 Illustration of non-constant surface slopes presented in a layer

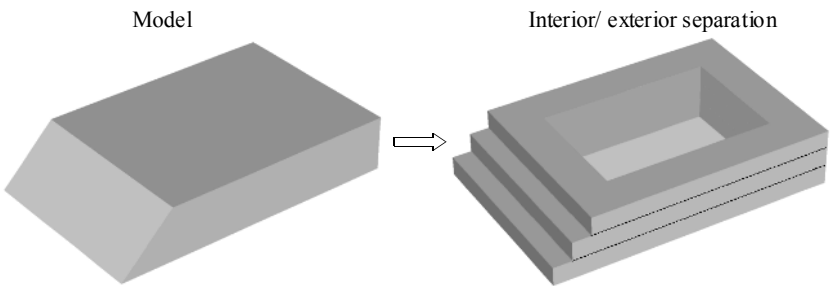
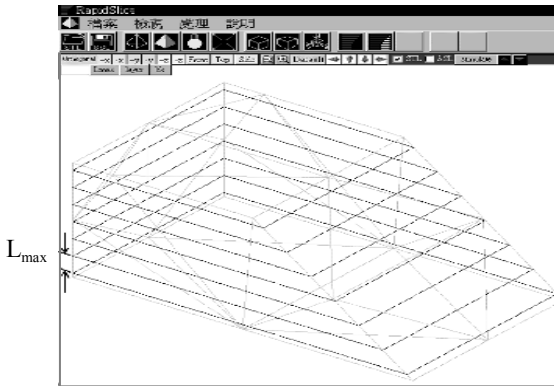


Fig. 5.15 Illustration of the fast interior and accurate exterior method

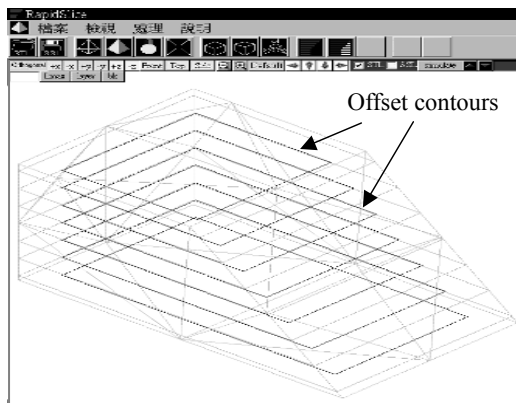
5.4.1 Procedures for advanced adaptive slicing

1. First of all, the part model is sliced by the machine allowable thickest layer ( $L_{max}$ ). The thickest-layer sliced result is shown in Fig. 5.16. This step saves the sliced contour information and the associated surface normal vectors, which will be used in step 3.



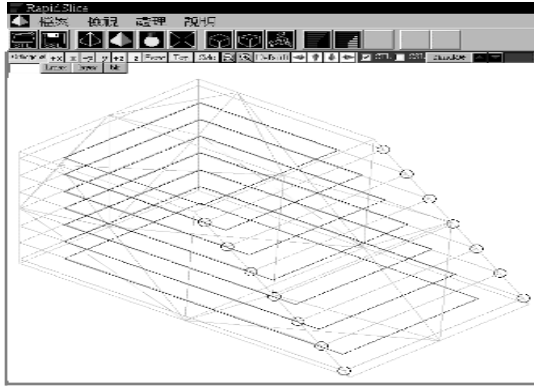
**Fig. 5.16** Contours sliced by thick layers

2. For each thick layer, the contour offset is calculated from the outside contour. The offset results are shown in Fig. 5.17. The area inside the offset contour will use the thickest layer deposition. The outside needs to be analysed to see if it needs further slicing; this is explained in the next step.



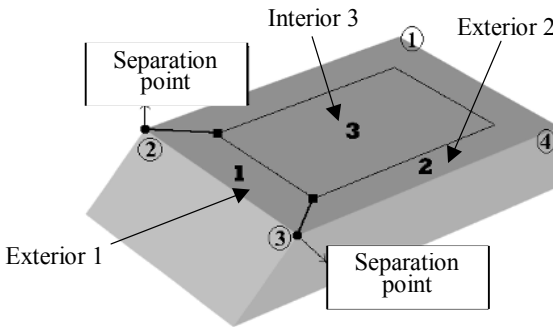
**Fig. 5.17** The offset contours for interior/exterior separation

3. The surface normal vectors, which were saved from step 1, are used to determine the separation points for the outside area. If the rate changes of the surface normal vectors of the outside boundary points are larger than a defined value, they are chosen as the separation points. The identified separation points, marked with circles, are shown in Fig. 5.18.



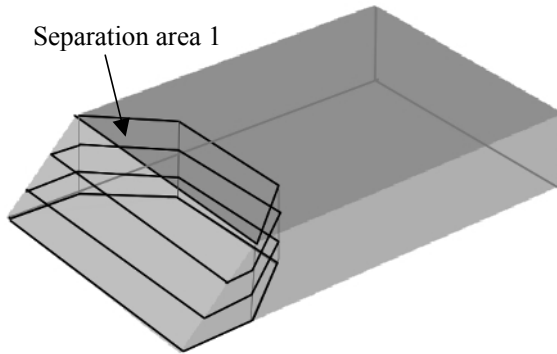
**Fig. 5.18 Separations determined on the boundary**

4. In the same layer, connecting the adjacent separation points with their offset points forms the separation area. The exterior will be separated into  $n$  areas if  $n$ -separation points are found in step 3. Figure 5.19 shows that one of the thick layers is separated into three areas: exterior 1, exterior 2, and interior 3. The boundary points 2 and 3 are identified as separation points for a single thick layer.



**Fig. 5.19 Three separated areas are identified for a thick layer**

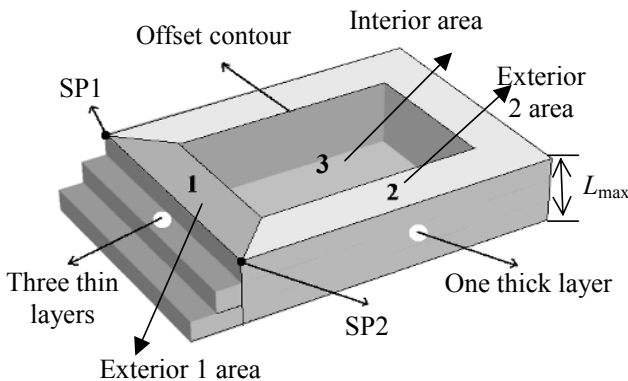
5. For each separated area, the basic adaptive slicing method is used to calculate the required layer number, called sub-layers. The sub-layer slicing result shows that the exterior 1 area requires three thin layers, the exterior 2 area requires only one thick layer, and the interior 3 area uses one thick layer.



**Fig. 5.20 Three sub-layers are needed for the first separation area**

6. The last step is to project the separation contour, which was formed in step 4, on to each sub-layer and form the new separation areas for each sub-layer. Figure 5.20 shows separation area 1 and three sliced sub-layers.
7. The sliced results for a thick layer ( $L_{\max}$ ) are shown in Fig. 5.21. To finish the part model slicing, steps 2 to 6 are repeated for all thick layers that were produced from step 1.

Using the above advanced adaptive slicing procedures for the demonstrated example, the fabrication sequences for one thick layer are: (a) three thin layers for exterior 1 area, (b) one thick layer for exterior 2 area, and (c) one thick layer for the interior area.



**Fig. 5.21 Advanced adaptive slicing results of a thick layer**

### 5.4.2 Analysis

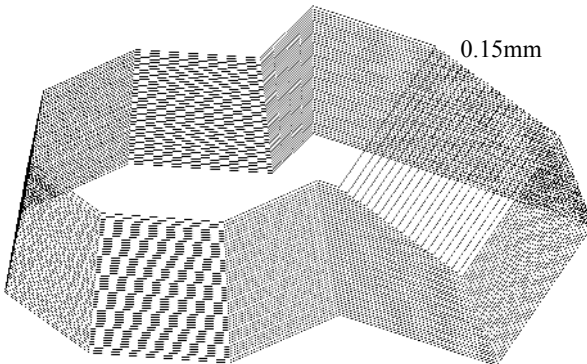
An example, as shown in Fig. 5.22, is used to demonstrate the results for different slicing algorithms, including constant thickness layer slicing, basic adaptive slicing, fast interior and accurate exterior slicing, and the advanced adaptive slicing methods. The user-defined allowable stair-stepping error is 0.08 mm. The assumption for adaptive slicing is that the RP machine can choose the varied layer thickness from 0.15, 0.2, 0.3, and 0.6 mm.

#### 5.4.2.1 *Constant thickness layer slicing method*

In order to keep the stair-stepping error under 0.08 mm, the constant layer thickness used for slicing is 0.15 mm. The sliced result is shown in Fig. 5.23.



**Fig. 5.22** A demonstrated example for different slicing algorithms



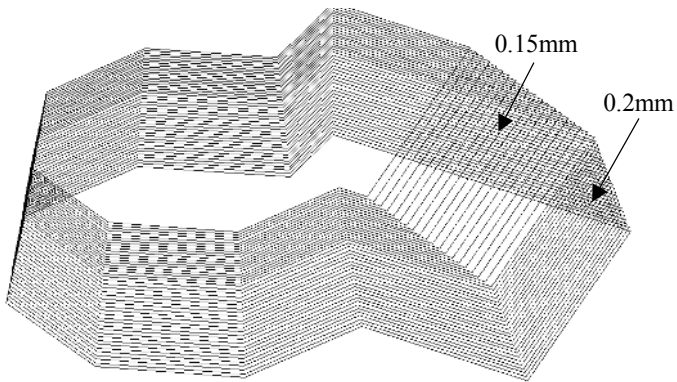
**Fig. 5.23** Slicing results using the constant thickness layer method

#### 5.4.2.2 Basic adaptive slicing method

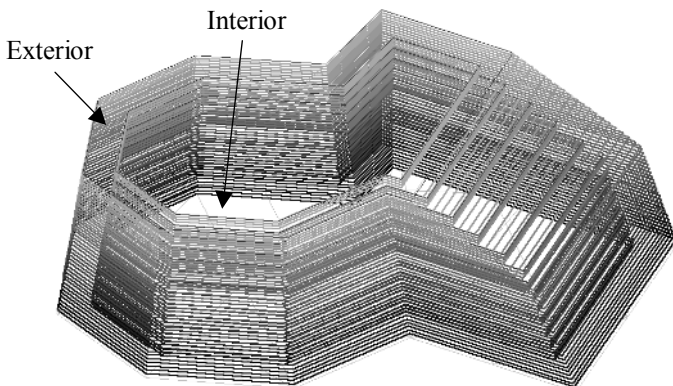
The layer thickness is determined by the changes of the surface slopes. In this case, two kinds of layer thickness are implemented to satisfy the surface accuracy requirement. According to the surface geometry changes, the upper portion uses 0.15 mm thickness layers, and the lower portion uses 0.2 mm thickness layers. Figure 5.24 shows the sliced results from the basic adaptive slicing method.

#### 5.4.2.3 Fast interior and accurate exterior method

The first step for this method is the exterior and interior separation. The offset to separate the exterior and interior is 1.5 mm. Figure 5.25



**Fig. 5.24** Sliced results using the basic adaptive slicing method



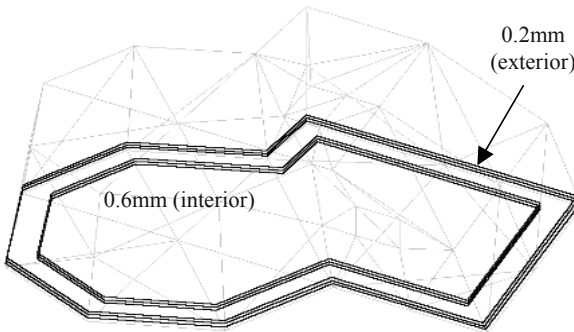
**Fig. 5.25** Sliced results for the thick interior and thin exterior method



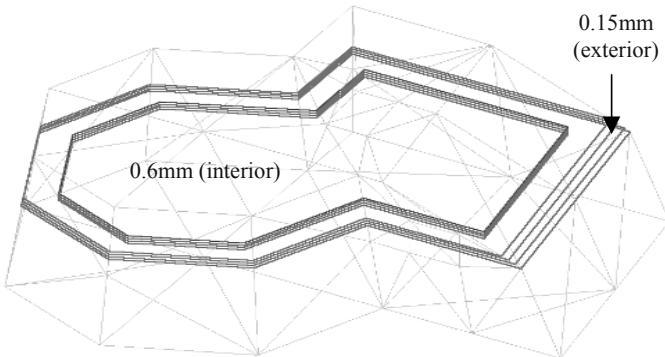
shows the sliced results with the exterior and interior separations. Figure 5.26 shows the exterior and interior areas of the first thick layer. The interior uses a thickest layer of 0.6 mm to speed up the fabrication process. The exterior uses 0.2 mm-thick layers to keep the stair-step error under 0.08 mm. The thinner layer, 0.15 mm, is needed for the exterior portion of the ninth thick layer, according to the surface geometry changes. Figure 5.27 shows the sliced results for the ninth thick layer where the interior uses one thick layer (0.6 mm), and four thin layers (0.15 mm) for exteriors.

#### 5.4.2.4 Advanced adaptive slicing method

The sliced results using the advanced adaptive slicing method are shown in Fig. 5.28. The exterior and interior separation results are the

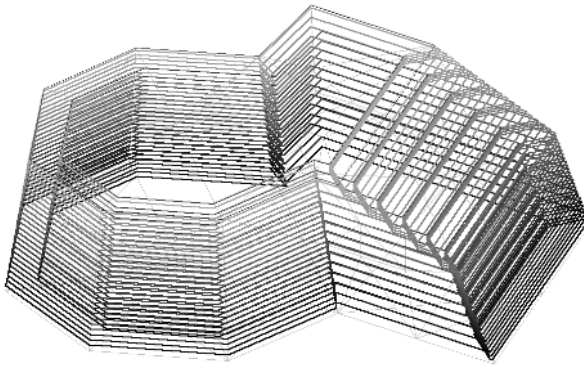


**Fig. 5.26** Exterior and interior areas of the first thick layer

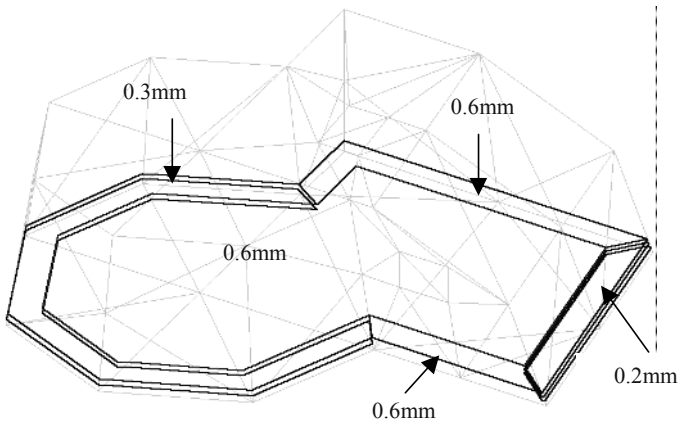


**Fig. 5.27** Exterior and interior areas for the ninth thick layer

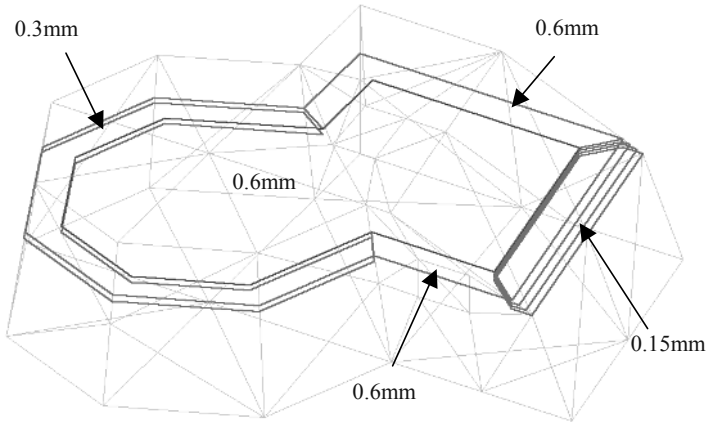
same as the fast interior and accurate exterior method. The difference is on the further separations in the exterior areas. For the first thick layer, the exterior area can be further separated into four segments, including two segments using one layer of 0.6 mm thickness, one segment using two layers of 0.3 mm thickness, and one segment using three layers of 0.2 mm thickness. Figure 5.29 shows the separation areas and segments for the first thick layer. Likewise, the exterior area for the ninth thick layer can be separated into 0.15, 0.3, and 0.6 mm layer segments, as shown in Fig. 5.30.



**Fig. 5.28 Sliced results for the advanced adaptive slicing method**



**Fig. 5.29 Sliced results of the exterior and interior areas for the first thick layer**



**Fig. 5.30 Sliced results of the exterior and interior areas for the ninth thick layer**

**Table 5.1 The comparison for different slicing algorithms**

Slicing algorithms	Constant layer	Basic adaptive	Fast interior/ accurate exterior	Advanced adaptive
Material deposition length (mm)	118 820	82 101	34 025	17 486
Relative fabrication time	100	69.1	28.6	14.7

Table 5.1 shows the comparison results for the different slicing algorithms. The first row shows the total material deposition lengths of the whole model fabrication for the different methods. The fabrication time calculation in the second row of the table is a relative value, compared with the constant layer slicing method. The fabrication time for the constant layer slicing is assigned to 100. The calculation assumptions are that the material deposition motion is under a constant speed and the time delay for layer changeover is neglected. The results for the relative fabrication time show that the constant thickness slicing method takes 7 times longer than the advanced adaptive slicing method for fabricating the demonstrated example.

## 5.5 CONCLUSION

Rapid prototyping fabricates the part layer by layer, which takes advantage of simple two-dimensional manufacturing instead of a complex three-dimensional process. The slicing operation is one of the major processes in CAD/CAM/RP systems; this transforms a three-dimensional part model into two-dimensional layers. Some of the RP research has focused on improving fabrication accuracy and efficiency. In general RP fabrication, the part accuracy depends upon the usage of layer thickness. The thinner the layer, the more accurate the part will be. However, the thin layer approach makes the RP fabrication inefficient.

Adaptive slicing techniques determine the minimum layer number required for RP parts under an allowable stair-stepping error; therefore, the fabrication time can be reduced. This chapter discussed the fundamental adaptive slicing algorithms as well as advanced adaptive slicing. The slicing results show that under the same surface accuracy condition the advanced adaptive slicing is 7 times faster than the conventional constant thickness layer processing, and twice as fast as the fast interior and accurate exterior method, for a demonstrated part model. The adaptive slicing technique takes into account the part surface geometry for slicing processes. This technique is particularly important for RP fabrication if the model contains fine features, and for micro RP fabrication.

## REFERENCES

- 1 Dolenc, A. and Mäkelä, I. (1994) Slicing procedures for layered manufacturing techniques. *Computer-Aided Design*, **26**, 119–126.
- 2 Kulkarni, P. and Dutta, D. (1995) Adaptive slicing for parametrizable surfaces for layered manufacturing. In *Proceedings of ASME Design Automation Conference*, Boston, Massachusetts, pp. 211–217.
- 3 Sabourin, E., Houser, S. A., and Bohn, J. H. (1996) Adaptive slicing using stepwise uniform refinement. *Rapid Prototyping Journal*, **2**, 20–26.
- 4 Sabourin, E., Houser, S. A., and Bohn, J. H. (1997) Accurate exterior, fast interior layered manufacturing. *Rapid Prototyping Journal*, **3**, 44–52.

- 5 Tata, K., Fadel, G., Bagchi, A., and Aziz, N. (1998) Efficient slicing for layered manufacturing. *Rapid Prototyping Journal*, **4**, 151–167.
- 6 Tyberg, J. and Bohn, J. H. (1998) Local adaptive slicing. *Rapid Prototyping Journal*, **4**, 118–127.
- 7 Hope, R. L., Jacobs, P. A., and Roth, R. N. (1997) Rapid prototyping with sloping surfaces. *Rapid Prototyping Journal*, **3**, 12–19.
- 8 Frank, D. and Fadel, G. M. (1995) Expert system based selection of the preferred direction of build for rapid prototyping processes. *Journal of Intelligent Manufacturing*, **6**, 334–339.
- 9 Cheng, W., Fuh, J. Y. H., Nee, A. Y. C., Wong, Y. S., Logh, H. T., and Miyazawa, T. (1995) Multi-objective optimization of part building orientation in stereolithography. *Rapid Prototyping Journal*, **1**, 12–23.
- 10 Lan, P.-T., Chou, S.-Y., Chen, L.-L., and Gemmill, D. (1997) Determining fabrication orientations for rapid prototyping with stereolithography apparatus. *Computer-Aided Design*, **29**, 53–62.
- 11 Majhi, J., Janardan, R., Smid, M., and Gupta, P. (1999) On some geometric optimization problems in layered manufacturing. *Computational Geometry: Theory and Applications*, **12**, 219–239.
- 12 Xu., F., Loh, H. T., and Wong, Y. S. (1999) Considerations and selection of optimal orientation for different rapid prototyping system. *Rapid Prototyping Journal*, **5**, 54–60.
- 13 Farouki, R., Konig, T., Tarabanis, K., Korein, J., and Batchelder, J. (1995) Path planning with offset curves for layer fabrication processes. *Journal of Manufacturing Systems*, **14**, 355–368.
- 14 Lin, G. S. (2000) Efficient advanced adaptive slicing method for layered manufacturing system. MS thesis, National Chung Cheng University, Taiwan.

# Chapter 6

## Multi-material representation and design issues

**G Fadel and S Morvan**

Mechanical Engineering Department, Clemson University, Clemson, South Carolina, USA

### ABSTRACT

This chapter discusses software representation issues as they pertain to the manufacturing of objects made of multiple materials on additive solid freeform fabrication machines. This capability has only recently become available, and it is only exploitable on certain machines. However, the potential of such a revolution in manufacturing is of such importance, that this chapter, and the following two chapters, are justified in a reference book on software issues for rapid prototyping.

To frame the topic, the chapter introduces this novel design and manufacturing paradigm, then presents past work at representing volumetric information in various fields. Finally, an approach to the representation based on a volumetric parametrization akin to the boundary representation (B-rep) extended to volumetric representation (V-rep) is detailed. Two other approaches are proposed in the following chapters.

### 6.1 INTRODUCTION

The design of mechanical components has always been presumably driven by their manufacturability, which is dictated by their geometry

or their shape. As CAD tools emerged, the manufacturing technologies remained the limiting factor to design freedom, and on a CAD system, the *de facto* representation for such an evolving object has always been its geometry. The CAD system provides this digital representation by means of mathematical formulations of the surfaces enclosing the object. From this representation, the location of the inside, the matter, is inferred, and its composition is assumed homogeneous.

It is only recently that this representation has been put under scrutiny as new manufacturing techniques have allowed a paradigm shift in the design process. It is now increasingly accepted that a design is more completely represented by its functionality rather than by its geometrical shape, which is viewed as a particular solution among many to achieve said functionality. Additionally, new manufacturing techniques such as rapid prototyping (RP) have helped increase design freedom by enabling the manufacture of components using additive methods, removing some of the limitations of traditional manufacturing.

The early methods for the fabrication of heterogeneous objects included powder metallurgy, physical and chemical vapour deposition, plasma spraying, self-propagating high temperature synthesis (SHS), and galvanofarming. A new breed of RP machines now allows parts made from multiple materials to be manufactured in either single or multiple processes. The introduction of multiple materials gives the added flexibility, and coincidentally the added complexity, of specifying the material composition at any location within the solid. Most conventional solid modelling schemes (CSG, B-rep, etc.) fail to capture information about the material distribution inside an envelope, as the task of representing solids has always been oversimplified to the representation of their enclosing boundary, which assumes a homogeneous material distribution inside this envelope.

As integration and miniaturization often push for parts to achieve multiple functionalities within a product, the design of solids made from multiple materials is appealing today but will eventually become a necessity. Akin to composites and carbon fibres, it allows the strength and features of a particular material to be used exactly where needed, thus evading the notion of trade-offs, which would normally have to be made when a particular material is selected. One subset of heterogeneous objects is known as functionally gradient materials

(FGM). These have the ability to exhibit continuously varying composition and/or microstructure, thus producing gradation in their properties.

One possible application for such a multi-material approach is the design of a turbine blade in which one might find the limited mechanical properties of a ceramic to be problematic in areas where mechanical strength would prevail over thermal properties. Conversely, in that same blade, the thermal properties of a titanium alloy might limit its usefulness in areas exposed to very high temperatures. Using a carefully designed mix of both ceramics and titanium throughout the blade may provide an adequate solution to this problem should these materials coexist in some manner.

As evidenced in this example, many applications based on the concept of functionally gradient materials or multi-materials can be developed to exploit the desirable material properties. In order to achieve mass applications of heterogeneous objects, systematic methodologies are needed. To date, the existing methods for the design and fabrication of heterogeneous objects tend to be experimental and *ad hoc*. Current research on heterogeneous object modelling has been primarily focused on representation schemes for heterogeneous objects. Therefore, the limited design tools available to obtain heterogeneous object models are only now emerging.

A new area of research, which will allow new representation schemes for these non-homogeneous objects, and whose evolution is dictated by a mix of traditional mechanical CAD and design automation tools is taking shape. Those representations should capture the peculiarities of a given manufacturing process while securing the essential information output by mathematical tools such as optimizers.

This chapter reviews traditional approaches to volumetric representations, and then focuses on one proposed alternative: volumetric-based parameterization. Chapters 7 and 8 present approaches based on constructive solid geometry (CSG).

## 6.2 LITERATURE REVIEW

In this section, a literature review of current heterogeneous solid representation efforts is presented. The task of representing



heterogeneous solids stems from the need to manufacture them, which in turn implies the ability to design them. Designing these objects can be done within a conventional design cycle (e.g. with a CAD system) used in conjunction with more sophisticated tools such as shape and material optimizers.

There is a distinction between the modelling of a heterogeneous solid and its representation. Modelling may seek to establish a finely grained model of the composite matter when it is subjected to various conditions (loads, thermal gradients, etc.), while the representation is concerned with the embodiment of the material composition within a mathematical formulation. The formulation proposed must be suitable for a variety of tasks that need to be performed on the solid herein represented (manufacture, simulation, design, etc.).

The representation of a continuously varying vector field of the material composition over a volume is a relatively novel need in computer-aided design. The main contributions in that area essentially focus on decomposing solids into subdomains within which a single material distribution function is defined as a function of the two- or three-dimensional co-ordinates. The other efforts presented here are also relevant; they show that the methods that might at first seem intuitive for representing heterogeneous solids are in fact difficult or cumbersome to implement. Conversely, other representation methods from other disciplines such as geosciences modelling introduce specific trends. The following sections summarize past efforts pertaining to the representation of heterogeneous solids. The scope of this literature review has been broadened in order to incorporate the resolution of similar problems in other disciplines where similar issues are encountered. The problem at hand is understood to be the representation of varying properties (continuous or discrete) over a given two- or three-dimensional field.

### **6.2.1 Colour representation and colour schemes**

The specification of colour (1) found in the desktop publishing industry provides some similarities with the specification of a material composition over a two-dimensional geometry. In effect, the cross-section of a multi-material solid can be seen as a colour image. The colours used in an image are projected on a colour space (2, 3) specific to the target application of the colour, and the colours are encoded in

a system suiting the task at hand. The following subsections present common colour spaces and colour representation methods currently in use.

#### **6.2.1.1 Colour spaces**

Whether colours are defined for a printer, a monitor, a film imager, or a high definition television, colours can be defined with respect to a device-independent standard based on the Commission Internationale de l'Eclairage (CIE) (4) standard for colour measurement. Given such a definition, the *gamut* is defined. The gamut is the extent of all possible colours reproducible by each device. Any device capable of reproducing colour, whether it be a colour printer, an analogue TV set, a computer colour monitor, or a liquid crystal display (LCD) monitor, exhibits a range of reproducible colours peculiar to the hardware used to render or display those colours. Colours falling outside of a particular device gamut cannot be reproduced. In the publishing industry, computer illustrations often exhibit a gamut broader than that of a printer and require adjustments to accommodate the limitations of the colour printer.

Whether the colours are used for video, photo applications, or magazine illustrations, appropriate colour spaces have been tailored in each case so that the creation medium becomes coupled with the reproduction medium (5). The profiling of a colour printer within a known colour domain ensures colour accuracy and fidelity, meaning that a colour output by a printer will closely match the tone of that same colour when it is displayed on a computer monitor. Unexpected results might occur in the event that some features are outside of the reproduction medium capabilities.

#### **6.2.1.2 Colour representations**

Colour is the perceptual result of light in the visible region of the spectrum, having wavelength in the region of 400 nm (red) to 700 nm (blue), incident upon the retina. The human retina has three types of photoreceptor cone cells, which respond to incident radiation with different spectral response curves (6). There is a fourth type of receptor in the eye, which only serves in low light situations.

Since there are only three types of colour receptors, three variables are sufficient to identify a colour. A colour is also an electromagnetic wave

that can be characterized using its spectral power distribution (SPD). A corresponding unique triplet specifying a colour can be obtained from an SPD using standardized methods created by CIE. These three numbers represent a colour in the CIE-XYZ colour representation (4). The CIE-XYZ colour representation is tied to the physics of the phenomenon, rather than to its visual appearance, and is somewhat impractical for use in digital art. To that effect, other colour representations have appeared, such as:

- RGB: a colour is specified in terms of the red, green, and blue primaries,
- CMY: a colour is specified in terms of the cyan, magenta, and yellow primaries

Typically, computer monitors exhibit phosphors in the three RGB primaries (red, green, and blue), while printers possess a set of toner cartridges resembling the CMY primaries (black is often added for efficiency; the representation is then termed CMYK).

### ***6.2.1.3 Colour spaces and representations summary***

A colour is a simple yet difficult phenomenon to represent uniformly, more so to reproduce. A representation alone is insufficient to fully characterize a colour it must also be accompanied by an environment that understands the purpose, or end use, of a particular colour before it can decide on a reproduction methodology.

It is evident that similar consideration should appear in the representation and the manufacturing of multi-material solids. A representation must be designed, along with the manufacturing environment, to couple the design tools and manufacturing means.

Nevertheless, in the case of the RGB representation, each colour is broken down into individual components representing the contribution of a primary colour to the original colour. The addition of all individual primary colours weighted with the components, causes the actual colour to appear (this is an additive representation). Again, heterogeneous materials are likely to face similar issues when a material with arbitrary characteristics (strength, thermal conductivity, colour, etc.) is created from the deposition of primary materials.

## 6.2.2 Geosciences modelling

Geosciences modelling shares some similitude with mechanical CAD, especially in the area of domain (two- or three-dimensional) representation – two-dimensional geographic domains can be represented by polygons, and a three-dimensional hill can be represented with a set of two-dimensional elevation maps. The added dimension of attributes such as soil composition, humidity, and erosion tends to make this modelling certainly very relevant for the problem at hand. The critical aspects relevant to us in geosciences modelling (7) are those of uncertainty and fuzziness of the boundaries of the objects described.

According to reference (8) there are two main geographic data models used: the exact model and the continuous field model (Figs 6.1a and 6.1b, respectively). The exact model is made of adjoining polygons tagged with a number of attributes representing a region with set properties (such as soil composition, pollution, etc.). Conversely, the continuous field model, instead of attaching the attributes to a topology, assumes that each attribute is a continuous and smoothly varying function over space. In practice, the function is often discretized on a regularly spaced grid and the overlay of all the grids should provide a meaning similar to that of the exact object model (with the exception of the boundaries).

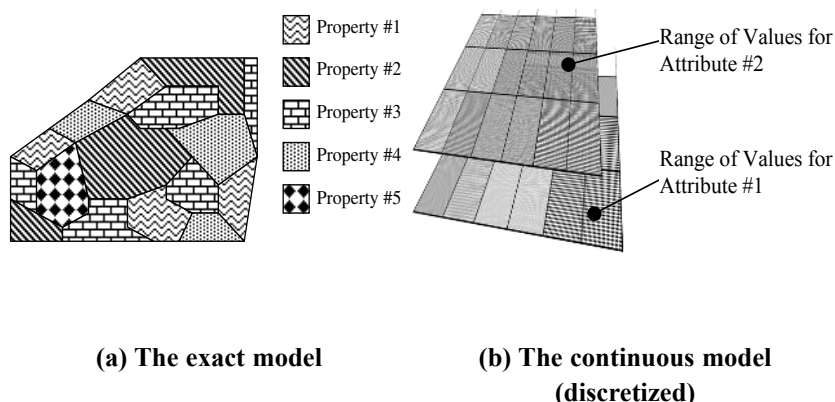


Fig. 6.1 Geographic data models

Strictly speaking, heterogeneous solids do not have any uncertainties associated with their boundary definition. Evidently, the designer of a part knows exactly where the material is, with the implicit assumption that it is located inside the boundary. It is interesting to observe how, for example, a soil boundary is modelled and how the change of a soil (which is often gradual) is treated in terms of fuzziness of the set (the uncertainty, due to measurement errors, is irrelevant here). For soils, the ‘core’ region is defined as the region that attains maximal membership in a given attribute, while the boundary is defined as the region attaining between zero and the maximal membership (the latter interval being open on both ends). The use of fuzzy logic to model the boundary is proposed by Ustry (9), and a similar approach, pertaining to a gradual change between two materials, is certainly an interesting issue. A more elaborate discussion to differentiate a ‘Boolean region’ from a ‘fuzzy region’ is presented in reference (8). This work illustrates why the problem of point membership in a region cannot systematically be answered by ‘neither true nor false’ in terrain modelling, since the soil composition can only be estimated rather than measured with absolute precision.

The two models described above are used to modulate and represent variable properties in a three-dimensional domain. How to represent this domain is the topic of the next section.

### **6.2.3 Volumetric representations (geometry)**

Volumetric representations can be very different. For instance, voxels represent an elemental formulation, the dimensionless voxel, the equivalent of a pixel in three-dimensional space. However, since discretizing a volume into dimensionless (or very small) elements can be overwhelming, methods to subdivide volumes have evolved. Regular shapes such as octrees (the volumetric equivalent of a quadtree) refine a volume into cubic elements of increasingly small size until the appropriate shape is captured. A more general space partitioning scheme is the cell tree, less used, but more flexible in spite of some limitations. These approaches are described next.

#### ***6.2.3.1 Solids composed of voxels***

Two definitions are commonly found for a voxel, the elementary unit of a volumetric representation.

- A voxel, like a pixel, has zero dimension. Akin to the way that a digital image is represented by an array of pixels, a volumetric dataset is made of voxels laid out on a regular three-dimensional grid. For every voxel in this dataset, a scalar value quantifies the membership of this voxel to a given reference material. Often, such a representation can be termed ‘spatial enumeration’.
- A voxel can also be considered to be a cube of small size.

Since the definition is not strict in order to remain consistent with the definition of a pixel, a voxel is assumed here to be a three-dimensional extension of a pixel and thus, has no dimension associated with it.

In some applications, the arrangement of voxels has been altered to permit a higher resolution in a given direction. Generally, volumetric data results from the assembling of cross-section images. Thus, the size of the image remains relatively fixed, while the spacing of the cross-sections is decreased to permit a more accurate reconstruction.

### **6.2.3.2 Volume rendering**

Traditionally reserved for medical imaging, volume rendering is now finding its way into areas such as failure analysis, computational fluid dynamics, meteorology, and terrain mining. This is a direct by-product of advancements in microprocessor architectures in desktop computers, the creation of embedded volume rendering hardware in custom graphics expansion boards, and their support in standard three-dimensional programming graphics libraries. Moderately complex sets (e.g. volumes featuring 512 voxels in  $x$ -,  $y$ - and  $z$ -directions) can be rendered, panned, and zoomed on ordinary computers in almost real time.

Although volume rendering **(10)** refers to the act of rendering volumetric data, the primary concern is not with this task, but rather with the underlying representations used for the data to be rendered, including but not limited to the algorithms involved in the handling of these data.

The principles of volume rendering are similar to those of ray tracing: for every pixel on the view plane, a ray of light is cast orthogonal to the view plane and directed toward the objects to be rendered. Each elementary element of volume (a voxel) intersecting the ray is composited (or combined) with other intersecting voxels to provide the

final intensity of the pixel on the view plane. This computation is performed repeatedly for every pixel on the rendered image, and for every view generated of the volume. It is sometimes convenient to attribute a weight to voxels of similar energy absorption, thus filtering out some tissues while emphasizing others. This classification task permits the separating of muscles and bones with the use of a transfer function, therefore emphasizing materials of similar energy absorption.

A noteworthy example using these techniques is the Visible Human (11) where a human body was entirely digitized through a similar process, resulting in about 1800 cross-sections of the body.

### 6.2.3.3 *Voxel-based representations*

The use of voxels to represent matter raises an interesting issue when it comes to rendering the outer surfaces of objects. These surfaces exhibit a shape that is often inappropriately captured by voxels. Nevertheless, these surfaces are rendered as if they were described by a conventional polygonal mesh using a normal estimation algorithm and an interpolation kernel (12).

Additionally, since the voxels are a sampling of physical data, the storage requirements for those sets are severe: a model containing a grid made from 512 arrays of  $512 \times 512$  voxels uses at the very least 128 Mbyte of memory (for 1 bit encoding). The dataset is usually very large and easily saturates most of a computer's memory bandwidth every time a new view is rendered. Volumetric datasets emanating from human organs are extremely dense, and much of the voxels forming the volume are non-empty. There is therefore little incentive to design space-efficient data structures. The main efforts made to improve the storage of volumetric datasets are aimed at improving the speed of volume rendering algorithms by pre-processing the voxels (13, 14) and skipping empty cells (15).

Another application of a voxel-based modelling scheme for solids is also presented in reference (16), where the aim is to see how a voxel-based modelling fits within an RP enabled manufacturing environment.

It should be noted that the use of voxels for solid modelling does greatly simplify CSG operations on solids (12). Boolean operations

performed on solids are reduced to plain voxel to voxel operations whose outcomes are extremely simple to compute. For instance, an empty voxel intersecting a full voxel is an empty voxel, while an empty voxel 'union' a full voxel is a full voxel.

#### 6.2.3.4 Octrees

Samet (17) proposed to use a hierarchical space-partitioning scheme in order to store volumetric data, specifically the octree. The principles of an octree are simple: to recursively subdivide a cube into eight smaller cells one-eighth the size of the original cube, until either the cube is empty or its content is below a set threshold size. Ayala *et al.* (18) proposed to perform Boolean operations and rigid body motions on octrees. Although octree encoding has interesting potential, it is in the termination criteria of the decomposition sequence that the complexity lies. Some example termination criteria, for three-dimensional polyhedra, are:

- A single vertex in the cell
- A single edge in the cell
- A single face contained in the cell

In the case of the octree two-dimensional counterpart, termed the quadtree, it is demonstrated that regardless of the termination criteria used, the number of cells is proportional to the perimeter of the object to be decomposed and to the resolution of the decomposition (17). Thus, for complex closed hollow two-dimensional polygons, a quadtree decomposition can yield a large number of cells, which restricts its use.

A solid modelling application using an octree-based representation scheme is presented in reference (19). In this application the termination criteria of the decomposition is to have a single face contained within each cell. Once again, a noteworthy benefit from this volume-based decomposition is the simplicity of Boolean operations on solids; though these are slightly more complicated than the operations performed on voxels (cells of different sizes may require additional treatment), their outcome is easily predicted. Also of importance, rigid body transformations (rotations, scaling, and translations in three dimensions) are also described without the application of a transformation matrix to the co-ordinates of the vertices of the objects and recreating a new decomposition; the



rotation is performed directly on the octree, as opposed to the geometry.

In fact, octrees and quadtrees are considered to be a special case of a more general space-partitioning scheme known as binary space partitioning (BSP). Instead of uniformly subdividing an element into either four or eight subelements (quadrees and octrees respectively), BSP subdivides an  $n$ -dimensional space using an  $(n-1)$ -dimensional hyperplane at an arbitrary location and orientation. When  $n = 2$ , e.g. in two dimensions, the hyperplane is a line, while when  $n = 3$ , e.g. in three dimensions, the hyperplane is a plane. The exact location and orientation of the subdivision is typically dependent on heuristics that seek to balance all nodes to minimize the depth of the tree. A comprehensive summary of BSP trees with implementation specifics is presented in reference (20). An overview of space-partitioning schemes is presented in reference (21).

#### 6.2.3.5 *The cell tree*

The cell tree (22) is an encoding for general polyhedral point sets of arbitrary dimensions (bounded or unbounded). This encoding represents a polyhedral by the algebraic sum of simpler, convex polyhedra (holes are 'subtracted'). Each convex polyhedral chain is described by the intersection of half-spaces and is represented by a vector. A half-space partitions an  $n$ -dimensional space in two halves, and is a hyperplane of dimension  $n-1$ . The representation is assumed to be minimal; a half-space not intersecting any other half-space (e.g. empty intersection) is removed from the vector.

Also, since the description is minimal, a half-space is a boundary of the convex polyhedral it is describing. To further normalize this representation, the half-spaces used in the description of a polyhedral are listed in a single location. A convex polyhedral is then represented by a list of 1, -1, or 0, referring to whether or not this polyhedral is, respectively, using a given half-space, its opposite, or not using it.

Performing Boolean operations on polyhedra is then a slightly more complicated matter than with the previous representation schemes, though it does not necessitate elaborated algorithms. The union of two polyhedra involves merging the two databases of half-spaces and adding more convex polyhedra to the chain. Subtracting a polyhedral

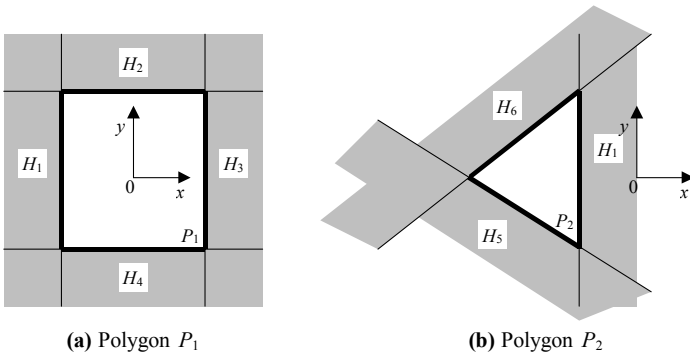
is a similar task, since the description allows unbounded polyhedra; the complementary of the polyhedral to be subtracted is created by negating the list of vectors it uses, and then adding that object to the convex polyhedra chain.

$P_1$  and  $P_2$  are written on the non-minimal basis formed by the hyperplanes  $H_i$ . Each convex polygon is written as a linear combination of those hyperplanes:

$$P_1 \equiv \sum \alpha_i \cdot H_i = 1 \cdot H_1 + 1 \cdot H_2 + 1 \cdot H_3 + 1 \cdot H_4 + 0 \cdot H_5 + 0 \cdot H_6$$

$$P_2 \equiv \sum \alpha_i \cdot H_i = -1 \cdot H_1 + 0 \cdot H_2 + 0 \cdot H_3 + 0 \cdot H_4 + 1 \cdot H_5 + 1 \cdot H_6$$

$P_1$  and  $P_2$  are shown in Figs 6.2a and 6.2b, respectively.



**Fig. 6.2 Cell tree decomposition of two convex polygons**

Both expressions can be rewritten in a vector form:

$$P_1 \equiv \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \\ H_5 \\ H_6 \end{bmatrix} = \mathbf{a}_1 \cdot \mathbf{H}$$

$$P_2 \equiv [-1 \ 0 \ 0 \ 0 \ 1 \ 1] \cdot \begin{bmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \\ H_5 \\ H_6 \end{bmatrix} = \mathbf{a}_2 \cdot H$$

The union of  $P_1$  and  $P_2$  is given (Fig. 6.3) as

$$P_3 = P_1 \cup P_2 \equiv (\mathbf{a}_1 \cup \mathbf{a}_2) \cdot H$$

The following table gives the result of the intersections  $H_i \cup H_j$ :

$H_i \cup H_j$	-1	0	1
-1	-1	-1	*
0	-1	0	1
1	*	1	1

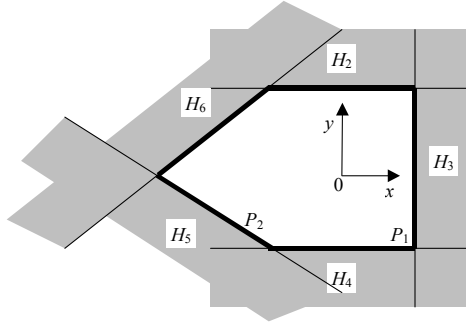
$P_3$  is then expressed as

$$P_3 \equiv [0 \ 1 \ 1 \ 1 \ 1 \ 1] \cdot H$$

It is interesting to note that this representation offers some features that other representations do not: in the case of three-dimensional polyhedra, there is no need to evaluate the vertices or the edges, strictly speaking, the entire shape is represented using only planes. Evidently, such a representation is permitted only on the convex class of shapes, making this a rather severe restriction. Though there are some known algorithms to decompose a three-dimensional solid by way of convex polyhedrons, this remains a difficult problem (23).

## 6.2.4 Interpolation and implicit methods

Having described methods to represent volumetric entities, the next topic focuses on means to formulate mathematical functions that can help represent such entities when starting from scattered data distributed over the domain. This section starts by looking at interpolation in three-dimensional space, and then focuses on implicit representations.



**Fig. 6.3 Union of  $P_1$  and  $P_2$**

#### 6.2.4.1 Three-dimensional interpolation from scattered data

The interpolation of scattered data is desired whenever the value of a physical quantity only known from points *sufficiently* populating the plane is sought at an arbitrary location  $(x, y)$ . The original (scattered) data points can be the result of computations or measurements that are otherwise difficult or impossible to reproduce. This interpolant can be used to provide derivatives, integrals, or visual representations of the underlying function.

Shepard-based methods feature local interpolants blended or modulated together (24). Once the degree of differentiability required is selected, the construction of the interpolating function involves the computation of unknown coefficients in the localized interpolant using the least squares method. The localized interpolating polynomials are then *blended* together by way of weight functions that are inversely proportional to the Euclidean distance from the centre of their respective support to the point under consideration.

##### *Interpolation principles*

Shepard's method constructs a *smooth* bivariate function  $f(x, y)$  interpolating data values  $f_k$  at scattered nodes defined by their coordinates  $(x_k, y_k)$ ,  $k = 1, 2, \dots, N$ :

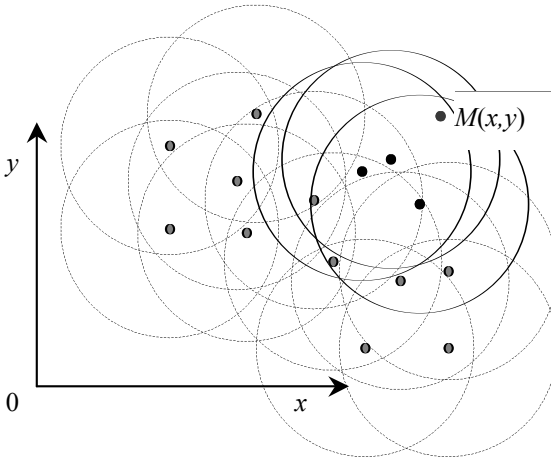
$$f(x, y) = \frac{\sum_{k=1}^N W_k(x, y) \cdot C_k(x, y)}{\sum_{k=1}^N W_k(x, y)}$$

The  $W_k(x,y)$  are un-normalized weight functions inversely proportional to the inverse distance function:

$$W_k(x,y) = \left[ \frac{(R_\omega^k - d_k)_+}{R_\omega^k \cdot d_k} \right]^p$$

where  $d_k(x,y)$  is the Euclidean distance between  $(x,y)$  and  $(x_k, y_k)$ ,  $R_\omega^k$  is the radius of influence (Fig. 6.4) about  $(x_k, y_k)$  that includes  $N_\omega$  nodes ( $N_\omega$  is fixed),  $p$  is an arbitrary exponent ( $p = 2, 3, \dots$ ), and

$$(R_\omega^k - d_k)_+ = \begin{cases} R_\omega^k - d_k, & d_k < R_\omega^k \\ 0, & d_k \geq R_\omega^k \end{cases}$$



**Fig. 6.4 Radii of influence for scattered points**

The  $C_k(x,y)$  values are selected according to the degree of differentiability required (the smoothness). They were originally introduced as bivariate polynomials of degree 2, and have been extended to polynomials of degree 3 and cosine series models in reference (25). For a polynomial-type interpolant of degree  $n$ :

$$C_k(x,y) = \sum_{i=0}^n \prod_{j=0}^i a_{i^2+j,k} \cdot (x-x_k)^j \cdot (y-y_k)^{i-j}$$

The computation of the indeterminate coefficients in  $C_k(x,y)$  is achieved using the least squares method to minimize the error function:

$$E = \sum_{\substack{i=1 \\ i \neq k}}^N E_k \quad \text{where} \quad E_k = \omega_{ik} \cdot [C_k(x_i, y_i) - f_k]^2$$

for

$$\omega_{ik} = \left[ \frac{(R_c^k - d_{ik})_+}{R_c^k \cdot d_{ik}} \right]^2$$

Again,  $R_c^k$  is the radius of influence about node  $(x_k, y_k)$  such that within  $R_c^k$  there are  $N_c$  nodes ( $N_c$  is fixed). The minimization of the error  $E$  is difficult and results in a large, sparse system of equations. As for the definition of the latter radius of influence, each  $E_k$  is coupled with  $N_c$  other  $E_i$ s such that the distance from  $(x_k, y_k)$  to  $(x_i, y_i)$  is less than  $R_c^k$ . Furthermore, at each node  $(x_k, y_k)$  the radius  $R_c^k$  must be recomputed, resulting in computational overhead. A triangulation of the plane, such as the Delaunay triangulation, helps simplify this process. A Delaunay triangulation of the nodes  $(x_k, y_k)$ ,  $k = 1, 2, \dots, N$ , is such that no nodes are contained in any of the formed triangle circumscribed circle.

Once the indeterminate coefficients in  $C_k(x,y)$  are computed, the computation of the  $W_k(x,y)$  requires finding  $R_c^k$ , the interpolation radius of influence about node  $(x_k, y_k)$ . The process of computing  $R_c^k$  is again simplified through the use of a Delaunay triangulation of the nodes.

### *Observations*

The Shepard method and its variants are suitable for the construction of an interpolating function over arbitrary arranged datasets, both in the plane and in space. A pre-processing step is however required, and additional data need to be stored for the evaluation of the interpolating function. The general-purpose nature of this method offsets its storage requirements and pre-processing stages, which are amortized over the evaluations of the interpolating function. The original aim of this algorithm is, however, not concerned with boundary value type problems (curves and/or closed portions of the plane), which is likely to be an issue for objects having exact boundaries.

It should be noted that the basic principles of Shepard-type methods (e.g. region of influence and distance-proportional weight functions) and some other variants are used to develop meshless techniques (26). Meshless methods are very promising analysis tools, especially in areas when traditional mesh-based finite element analysis (FEA) modelling is difficult to implement; e.g. when the medium modelled is undergoing important topological changes (crack propagation, forging, etc.). The domain *discretization* found in meshless techniques follows on from where the Shepard method left off: at the modelling of the boundary discontinuities in the weighting functions (27). Meshless methods do not have the same maturity level of traditional FEA (mesh-based) techniques, and the assembling of the stiffness matrix is in general much more complex and more computationally intensive. This computational overhead over traditional mesh-based techniques restricts its use to areas where mesh-based methods cannot be applied. Both meshless and mesh-based methods can, however, be used concurrently in a mixed analysis environment.

#### 6.2.4.2 Implicit representation of point sets

The implicit representation of point sets by the zeros of some real-valued function allows a greater freedom in the representation of surfaces (28, 29). Transfinite interpolation does not suffer from the same limitations because the surface is defined in implicit form; for example, in two dimensions let

$$f_1(x, y) = 2 - |x - y| - |x + y| \quad \text{and} \quad C_1 : \{(x, y) \in \mathbb{R}^2 / f_1(x, y) = 0\}$$

and

$$f_2(x, y) = 1 - x^2 + y^2 \quad \text{and} \quad C_2 : \{(x, y) \in \mathbb{R}^2 / f_2(x, y) = 0\}$$

$C_1$  and  $C_2$  are respectively defined as the set of points in  $\mathbb{R}^2$  solution of  $f_1(\mathbf{x}) = 0$  and  $f_2(\mathbf{x}) = 0$ .  $C_1$  and  $C_2$  are plotted in Fig. 6.5.

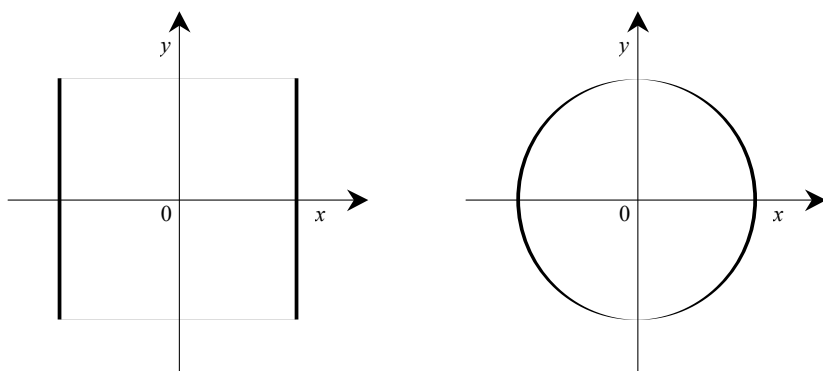
The set of points solution of  $f_1(\mathbf{x}) = 0$  is a square of dimension  $2 \times 2$ , centred around the origin, while  $f_2(\mathbf{x}) = 0$  corresponds to the unit circle. The solution of  $f_3(\mathbf{x}) = f_1(\mathbf{x}) \cdot f_2(\mathbf{x}) = 0$  is comprised of both the unit square and the unit circle centred on the origin. Therefore the unit square and the unit circle are both defined in a single equation,  $f_3(\mathbf{x}) = 0$

and although the two curves are disjoint, only one equation is sufficient to represent both sets of points (Fig. 6.6).

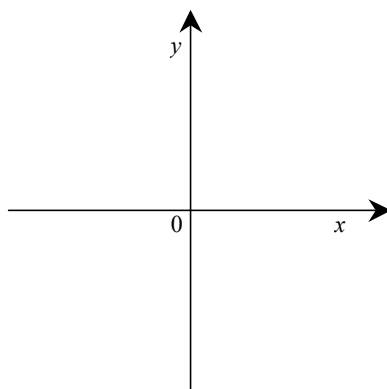
Based on those principles, and through the use of R-functions, implicit surfaces can be used to perform transfinite interpolation. The next section presents an overview of R-functions.

*Transfinite interpolation over implicitly defined sets*

A transfinite interpolation seeks to construct a function  $f(x)$  taking set values and/or derivatives over a collection of point sets. The latter set of points can be composed of isolated points, curves, or surface regions. Transfinite interpolation can be seen as the construction of a solution to a boundary value problem.



**Fig. 6.5** Set of points solutions to  $f_1(x) = 0$  and  $f_2(x) = 0$



**Fig. 6.6** Set of points solution to  $f_3(x) = 0$



Traditional interpolation techniques seek a surface passing through a network of curves, where both the surface and the curves are defined in parametric form. The parametric form of the interpolant restricts the topology of this network of curves, the continuity, and the smoothness of each individual curve. Such a surface is not permitted to be disjointed or to contain voids or holes. Typically a surface is created using a collection of patches, which are themselves created using refined surface interpolation techniques.

*Transfinite interpolation using R-functions*

The use of  $R$ -functions to describe geometric domains is explored in reference (30) and is done as follows. A function  $f(\mathbf{x})$  is constructed, taking on predefined  $f_i$  values at points  $\mathbf{x}_i$  such that

$$f(\mathbf{x}) = \sum_{i=1}^n f_i \cdot W_i(\mathbf{x})$$

Each weight function  $W_i(\mathbf{x})$  is inversely proportional to the distance from the point  $\mathbf{x}$  to the node  $\mathbf{x}_i$ . The weight functions  $W_i(\mathbf{x})$  are positive, continuous functions satisfying the interpolation condition  $W_i(\mathbf{x}_j) = \delta_{ij}$  and forming a partition of unity, i.e.

$$\sum W_i(\mathbf{x}) = 1$$

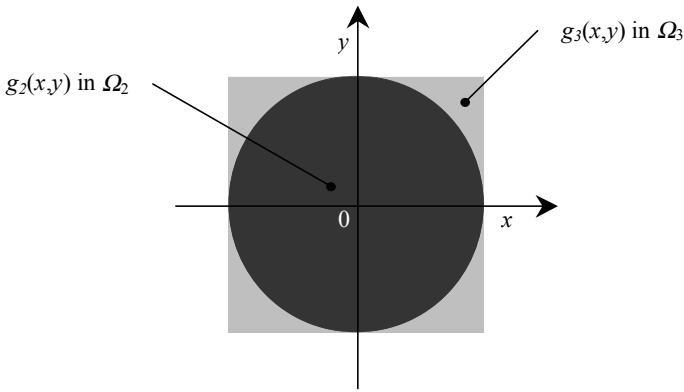
$W_i(\mathbf{x})$  can be built by normalizing the Euclidean distance from point  $\mathbf{x}$  to the node  $\mathbf{x}_i$ ,  $d_i(\mathbf{x})$ :

$$W_i(\mathbf{x}) = \frac{d_i^{-\mu_i}(\mathbf{x})}{\sum_{j=1}^n d_j^{-\mu_j}(\mathbf{x})}$$

The exponent  $\mu_i$  controls the behaviour of the interpolating function at the nodes, and the selection of the values of  $\mu_i$  permits the interpolant to be differentiable at  $\mathbf{x}_i$ . To improve numerical stability, the weight functions can be rewritten as

$$W_i(\mathbf{x}) = \frac{\prod_{j=1, j \neq i}^n d_i^{\mu_i}(\mathbf{x})}{\sum_{k=1}^n \left( \prod_{j=1, j \neq k}^n d_j^{\mu_j}(\mathbf{x}) \right)}$$

Establishing the distance function  $d_i(\mathbf{x})$  is easily achieved for domains made from scattered points whereas it is more problematic for a curve or a closed domain such as a disc. For a single point, a distance function can be seen as an implicit function having a single point  $\mathbf{x}_i$  satisfying  $d_i(\mathbf{x}) = 0$ . Based on this observation, it is proposed to replace the distance function for every set by a normalized form of their implicit functions. Using the previous formulation for the unit circle and square, and assigning an arbitrary function to take value only on the domain defined by  $\Omega_3$ , while another function takes another value over  $\Omega_2$



**Fig. 6.7 Prescribed values over the domains  $\Omega_2$  and  $\Omega_3$**

The surface meeting the requirements illustrated in Fig. 6.7 is computed by using the following weight functions, and choosing  $\mu_2$  and  $\mu_3$  as 2:

$$W_2(x, y) = \frac{[f_3(x, y)]^2}{[f_2(x, y)]^2 + [f_3(x, y)]^2}$$

$$W_3(x, y) = \frac{[f_2(x, y)]^2}{[f_2(x, y)]^2 + [f_3(x, y)]^2}$$

The interpolating surface created has the following equation:

$$f(\mathbf{x}) = \sum_{2,3} f_i \cdot W_i(\mathbf{x}) = g_2(x, y) \cdot W_2(x, y) + g_3(x, y) \cdot W_3(x, y)$$

A sample plot of this three-dimensional surface, for  $g_2 = 1$  and  $g_3 = 2$  is shown in Fig. 6.8. Another plot of the same function  $f(x)$ , with the functions for  $g_2$  and  $g_3$  indicated below, is shown in Fig. 6.9.

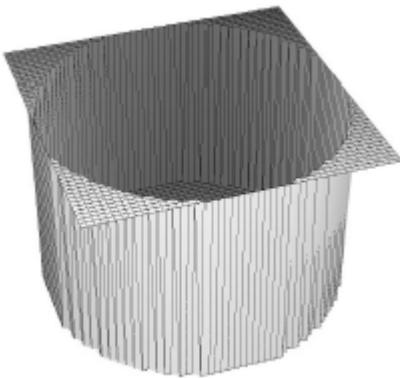
$$\begin{cases} g_2(x, y) = 3 - e^{-(x^2 + y^2)} \cos \left[ 5 \cdot (x^2 + y^2) \right] \\ g_3(x, y) = 1 + e^{-4[|x| - x_0|^w - |y| - y_0|^w]} \\ x_0 = y_0 = \frac{1}{2} \cdot (1 + 1 \cdot \frac{1}{\sqrt{2}}) \end{cases}$$

$g_2(x, y)$  provides undulations on the inner part of the surface, while  $g_3(x, y)$  provides ‘dimples’ on the four corners of the domain defined by  $\Omega_3$ .

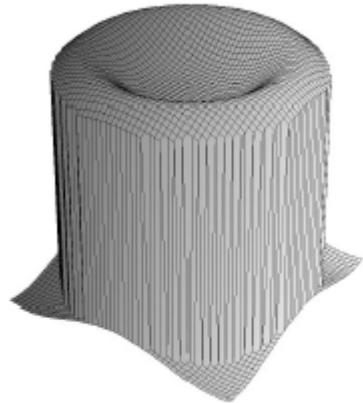
#### *R-functions*

*R-functions* are a class of functions permitting Boolean algebra to be achieved using real numbers (31). *R-functions* are built on the assertion that a positive number is considered to be ‘true’, and conversely, that a negative number is considered to be ‘false’.

An *R-function* has the essential property that its sign, and indirectly, its ‘trueness’ or ‘falseness’, can be determined from the signs of each of its arguments, regardless of their magnitude.



**Fig. 6.8** Sample plot for  $f(x)$



**Fig. 6.9** Another sample plot for  $f(x)$

Although  $R$ -functions may seem cumbersome at first, they bring a much-needed quality to any domain represented by a Boolean predicate, i.e. differentiability. When the parameter function  $\alpha(x_1, x_2)$  is chosen appropriately, both the ‘and’ and ‘or’ operators become differentiable at most points, thus guaranteeing smoothness of the domains built from these operators.

A geometric domain can be represented using several inequalities that have to be met concurrently (‘and’) or alternatively (‘or’). This set of inequalities can be represented with a single implicit function using  $R$ -functions in lieu of Boolean operators.

The theory of  $R$ -functions simplifies the definition of set theory modelling by providing a companion  $R$ -function to every logical set operator. These companion functions are differentiable and can be evaluated at any point of the plane.

### *Conclusion*

The transfinite interpolation technique has great potential, and the control of the smoothness of  $R$ -functions permits differentiability almost everywhere. The handling of discontinuities, caused by the fact that ‘0’ is neither negative nor positive is, however, more difficult. In most cases, the discontinuity appears at the junction between two domains and the interpolant needs to be carefully treated to prevent such problems. Furthermore, it is in general difficult to find an implicit function for domains enclosed by NURBS or spline curves, thus making this technique difficult to apply to general-purpose problems.

#### **6.2.4.3 Hypervolume interpolation**

Interpolating a solid with respect to its material properties, in the context of composite materials, has been addressed in reference (32). A hypervolume interpolation scheme is devised, with the construction of an interpolating function from a set of conditions/constraints imposed by the geometry and the material distribution. This approach extends surface interpolation techniques such as the Coons bicubic model to a tricubic representation. A detailed geometry model is constructed from the piecewise interpolation of smaller entities combined together, with continuity conditions occurring along each of the common surfaces of adjacent hyperpatches. Each hyperpatch interpolates the geometry through 64 suitably chosen points (Fig. 6.10).

This scheme is then explored for FEA, and the use of this tricubic interpolation along with a 64-point isoparametric element (using the same interpolation) is also studied, yielding exact integrals in the stiffness matrix (e.g. polynomials). Those integrals can be computed readily beforehand, and although the number of different coefficients is rather large (1728), being able to perform an analysis eliminating the error due to the stiffness matrix approximation is certainly appealing.

### 6.2.5 Previous work in multi-material/heterogeneous solid representation

Many representation schemes have been developed to represent solids. Manifold solids and  $R$ -sets were first proposed to represent solid models (33, 34). For conventional feature modelling, the use of a non-manifold structure was first proposed by Pratt (35). Selected geometric complex (SGC) is a non-regularized non-homogeneous point set represented through enumeration as the union of mutually disjoint connected open cells (36). Constructive non-regularized geometry (CNRG) was also proposed to support dimensionally non-homogeneous, non-closed point sets with internal structures (37). Middleditch *et al.* (38) present the mathematics and a formal specification for the mixed dimensional cellular geometric modelling.

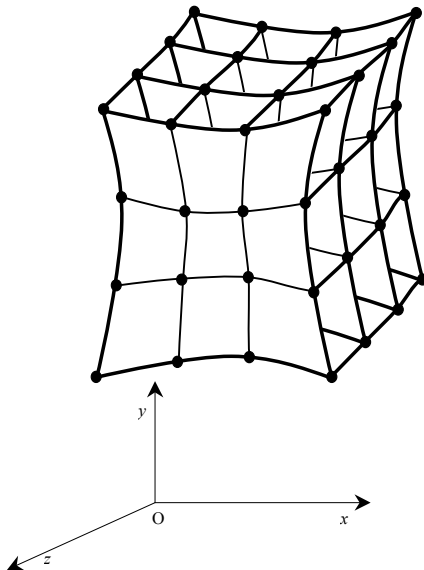


Fig. 6.10 The 64-node hyperpatch

Current research on heterogeneous objects has led to many representation schemes for heterogeneous object modelling. Kumar and Dutta (37) proposed  $R/m$  sets be used for representing heterogeneous objects. Jackson *et al.* (39) proposed another modelling approach based on subdividing the solid model into subregions and associating the analytical composition blending function with each region. Some other modelling and representation schemes, such as utilizing voxel models, implicit functions, and texturing, have also been proposed (30, 40–42). Some of these approaches are detailed below.

### 6.2.5.1 *Heterogeneous objects approach at the University of Michigan*

#### *Representing multi-material objects*

Initially, the efforts of the University of Michigan were aimed at establishing the theoretical grounds required to represent heterogeneous solids. In the work of Kumar and Dutta (43) an intuitive extension of single-material solids representations to multi-materials is presented. The presentation of the notations and the regularized CSG operators are thoroughly covered, along with an example implementation using the ACIS (44) solid modelling kernel.

Furthermore, the theory introduced earlier is applied and extended to gradient solids in another effort (43). A three-dimensional multi-material solid with gradient changes is a solid whose material distribution is location dependent in the volume under consideration. The material distribution is gradual in the sense that between two close locations, the material composition varies or remains constant. The material space  $T = \mathfrak{R}^3 \times \mathfrak{R}^n$  is introduced,  $n$  being the number of constituent materials.  $\mathfrak{R}^3$  is the geometrical space and  $\mathfrak{R}^n$  the material space with each dimension representing a primary material. The material composition at any arbitrary location is identified by volume fractions of each of the primary materials. The volume fraction is the set value of a basic material, or the value of a material component. For simplicity, all volume fractions are valued between 0 and 1. Void, or no material, is considered to be another material. Thus, the volume fraction must sum to 1, and the space of volume fraction is defined as

$$V = \left\{ \mathbf{v} \in \mathfrak{R}^n \mid \|\mathbf{v}\|_1 = \sum_{i=1}^n v_i = 1, v_i \geq 0 \right\}$$

Each point in an object  $S$  is then characterized in  $T$  as  $[\mathbf{x}, v(\mathbf{x})]$ , where  $\mathbf{x} \in S$  is a point in the object and  $v(\mathbf{x})$  the material at that point, such that  $v(\mathbf{x}) \in V$ . The paper then introduces several other notations, and of interest is the introduction of the mapping function  $F$  from the geometric points  $\mathbf{x}$  to the material space  $V$ . In practice, it might be difficult to use a single function  $F$  for each  $v_i(\mathbf{x})$  to characterize the entire object. Thus,  $F$  can be a piecewise function composed of a finite number of  $C^\infty$  functions. Though the details are somewhat vague about the actual implementation or choice of those functions, the example of a simplified valve seat is presented. The overall geometry is that of a solid of revolution comprised of three different parts. The outer shape of the valve seat is made of one material, Al33, while the inner part is made of brass. There is a transition zone where the material gradually changes from Al33 to brass. The piecewise construction of  $F$  is done over three different domains: the outer part, the inner part, and the transition zone. In this particular case, the simple geometry of revolution allows for such treatment while in other more complicated situations the construction of  $F$  may require several domains to be considered, regardless of whether or not they represent individual features of the part.

Finally, the issues of manufacturing are also covered, with the discretization of the distribution function at the interface, which is studied with respect to issues of manufacturing. Unfortunately, at the time of the article (1997) the authors did not have access to a multi-material machine, thus the article fell short of carrying the research from modelling through to manufacturing.

#### *Modelling and representation of multi-material objects*

In reference (38) the same theoretical background is re-introduced, and taken a step further with material considerations at the microstructure level. A one-dimensional material variation is considered with the power law (45):

$$F = v(\bar{\mathbf{x}}) = \begin{cases} (\alpha \cdot \bar{\mathbf{x}}^p) \\ 1 - (\alpha \cdot \bar{\mathbf{x}}^p) \end{cases}, \quad \alpha \geq 1$$

$\bar{\mathbf{x}}$  = normalized parameter with respect to the transition zone

$\alpha$  = scaling factor

$p$  = distribution exponent, varying from 0.1 to 10

More recently (38) and with the use of a direct material deposition (DMD) apparatus, the entire design process was considered from design to manufacturing while including design homogenization to obtain an ideal microstructure pattern distribution. The material considerations are at the microstructure level, which translates into large amounts of data and many variables to consider during the optimization stage. This representation lacks the ability to group together a large number of similar microstructures to reduce the amount of data needed. Manufacturing is confined to the presentation of the DMD apparatus and its operating principles. DMD is presented as a very potent alternative for the manufacturing of parts made from multiple materials and enables compositions to be varied arbitrarily.

#### *Unifying layered manufacturing data*

Another effort from the University of Michigan is aimed at the manufacturing of heterogeneous objects (46). There, the grounds for the unification of a layer file format are laid out. The impetus is to provide a machine-independent representation of the data generated during manufacturing of objects on rapid prototyping apparatus. The approach is then further extended and applied to the manufacturing of a heterogeneous object on a DMD apparatus.

#### *Standard for the exchange of product model data (STEP/ISO 10303)*

STEP is an ISO Standard (ISO 10303) aimed at facilitating the representation and exchange of digital product information(47, 48). Each aspect of ISO 10303 covers a particular aspect of the information to be represented. There are constant additions made to all aspects of this standard, all of which fall into the following categories, and in some instances, subcategories.

- Description Methods (e.g. Overview and Fundamental Principles #1, EXPRESS language ref. Manual #11, etc.)
- Application Protocols and Associated Test Suites (e.g. Mechanical Design Using Boundary Representation #204, Mechanical Design using Surface Representation #205, etc.)
- Integrated Information Resources, composed of several sub-items:
  - Application Modules (e.g. Colour #1002, Curve Appearance #1003, Shape Appearance and Layers #1009, etc.)
  - Integrated Application Resources (e.g. Ship Structures #102, Finite Element Analysis #104, etc.)
  - Integrated Generic Resources (e.g. Fundamentals of Product Description #41, Geometry and Topology Representation #42, etc.)



- Application Interpreted Constructs (e.g. Geometry Bounded Surface, Elementary B-Rep #513, Advanced B-Rep #514, etc.)
- Implementation Methods (e.g. Clear Text Encoding Exchange Structures #21, C++ Language Binding #23, FORTRAN Language Binding #25, etc.)
- Conformance Testing Methodology (e.g. Requirements on Testing Labs and Clients #32, etc.)

The STEP project is very complex and it will remain a work in progress for many years to come. Yet there is no current provision in STEP for addressing the problem at hand. Building on his work presented earlier, Dutta proposed an extension to STEP for the representation of heterogeneous solids (49), which followed from his previous attempt at extending STEP to include layered manufacturing (38).

In this research, a solid is considered to be an assembly of sub-components, each of which possibly being a heterogeneous solid. Dutta's representation decomposes heterogeneous solids into sub-regions where a material distribution function is defined in closed form over the subdomain. Thus, the task of representing heterogeneous solids based on Dutta's method is essentially limited to the representation of each subdomain for which a material transition function needs to be defined. As the representation of geometries is readily available in STEP, representing the subdomains was not problematic and each subdomain must be accompanied by a material distribution function. The definition of the material distribution function is permitted through the Mathematical Constructs defined in ISO 10303-50 (subsection of Integrated Generic Resources). The creation of local co-ordinate systems that can be attached to various geometrical entities (points, curves, surfaces, etc.) is allowed within another subsection of STEP which when combined with ISO 10303-45 (Materials) permits the complete definition of the material distribution function.

Arguably, translating a particular heterogeneous representation in STEP/ISO 10303 does not make this representation better than another. This does, however, validate the extension principles (e.g. building upon conventional solid modelling techniques) used to elaborate the theoretical aspect of this representation, which is nevertheless encouraging both for the development of STEP/ISO 10303 and for the development of a representation for multi-material solids.

### *Summary*

In summary, the work done at the University of Michigan lays down the general principles and addresses salient points of heterogeneous objects modelling:

- The composition at any location is specified using volume fractions.
- Void is considered to be another ‘base’ material.
- The blending functions, modelling gradual changes, will need to be described over non-intersecting subdomains. Some subdomains will have to be introduced in the representation for this sole purpose (making the representation more complex).

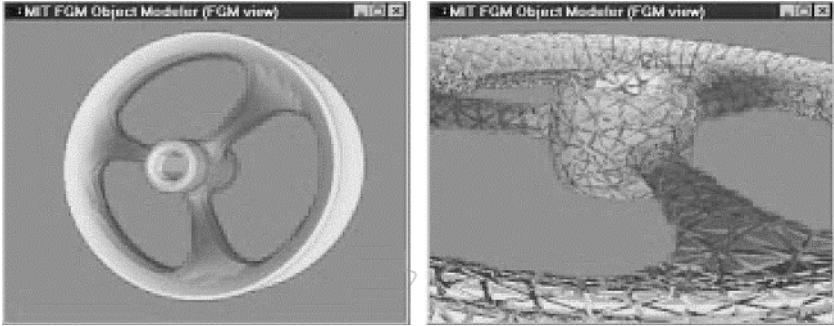
Most of this work on the modelling of heterogeneous objects was summarized in reference (50). Further extensions of this work are detailed in Chapters 7 and 8 of this book.

#### **6.2.5.2 Functionally graded material (FGM) modeller**

In references (51) and (52) a modeller (FGM modeller) incorporating a representation of heterogeneous solids is presented. A solid is decomposed into a tetrahedral mesh (Fig. 6.11) where the material is interpolated between the nodes (the material compositions is specified at the nodes). The interpolation of the composition is based on Bernstein polynomials, so the degree of the interpolation can be varied.

Decomposing a solid into a tetrahedral mesh is achieved using a finite element pre-processor providing a finite element mesh. The mesh is then imported in the FGM modeller, which incorporates a model processor unit that translates the FGM information into machine instructions for the rapid prototyping apparatus.

Manufacturing is achieved on a three-dimensional printing apparatus (53). Based on that experience, the authors conclude that design rules should be implemented in order to make the manufacturing process as robust as possible. One such rule would constrain the volume fractions of the constituent materials. They submit a lower and an upper bound on the volume fraction for each material. For instance, the binder’s lower limitation is imposed by the necessity of forming a solid part, which requires a minimal amount of binder to be deposited inside the model.



**Fig. 6.11 Decomposition of a solid in cells (51)**  
(see also colour plate section)

Additionally, in reference (39), the problem is presented as follows:

$$S \equiv C = \bigcup_{\kappa} c_{\kappa}$$

$C$  is a decomposition of a solid  $S$  using a cell-tuple structure representation (54). The cell-tuple structure proposes a representation based on an arrangement of cells (the  $c_{\kappa}$ 's), each representing a topological feature of the model (such as a vertex, edge, face, or region). In this representation, the entities are used in decreasing order of dimensionality: a region is bounded by several faces, a face by several edges, and so forth. In each  $c_{\kappa}$ , a composition function  $\mathbf{m}(\mathbf{x}) = [v_0(\mathbf{x}), v_1(\mathbf{x}), \dots, v_{n-1}(\mathbf{x}), v_n(\mathbf{x})] \in M$  where  $M$ , the material space, is defined. Assuming the composition is specified at a number of control points  $\mathbf{x}_{\kappa,j}$ , the problem is as follows for every cell  $c_{\kappa}$ .

Find an interpolation  $[\mathbf{x}_{\kappa}(u, v, w, t), \mathbf{m}_{\kappa}(u, v, w, t)]$  such that  $[\mathbf{x}_{\kappa}(u_j, v_j, w_j, t_j), \mathbf{m}_{\kappa}(u_j, v_j, w_j, t_j)] = [\mathbf{x}_{\kappa,j}, \mathbf{m}_{\kappa,j}]$  at the control points. The barycentric Bernstein polynomials  $B_i^n(u, v, w, t)$  are used to blend the control compositions:

$$[\mathbf{x}_{\kappa}(u, v, w, t), \mathbf{m}_{\kappa}(u, v, w, t)] = \left[ \sum_j B_i^{n_{\kappa}}(u, v, w, t) \cdot \mathbf{x}_{\kappa,j}, \sum_j B_i^{n_{\kappa}}(u, v, w, t) \cdot \mathbf{m}_{\kappa,j} \right]$$

$$B_i^n(u, v, w, t) = \frac{n!}{i!j!k!l!} u^i v^j w^k t^l$$

$n_g$  and  $n_m$  being the degrees of variation in shape and composition. It is unclear, though essential, if this particular interpolation of the material composition enforces the material ‘completeness’ constraints, e.g.  $|\mathbf{m}_e(u, v, w, t)| = 1$  anywhere within a solid represented so.

Two classes of design rules that ought to be enforced to facilitate manufacturing are briefly proposed: one design rule limiting maximum and minimum composition and another one governing the rate of change in material composition. Finally, it is mentioned that the decomposition, originally chosen arbitrarily from the material distribution, should be dependent and eventually refined to accommodate areas of high or low rates of change in material. Areas of high rates of change should be subdivided, and conversely, areas of low rates of change should be grouped together to reduce complexity, akin to a decimation algorithm used on triangular meshes. Finally, means to design the composition globally are discussed. This particular global composition is set as a function of the distance from a geometric feature, such as an axis, a reference plane, a surface, or a point to the location under consideration. The practical implementation of such a distance function is discussed, along with efficiency considerations.

### **6.2.5.3 Summary of heterogeneous solid modelling representations**

There is a growing interest in the modelling of heterogeneous objects. The enabling technologies behind the creation of these objects, structural optimization and multi-material manufacturing, are maturing and adding an increased tangibility to this field.

Dutta’s modelling scheme draws on the same approach as that of surface modelling extended to the modelling of materials as a function of geometric variables. The original impetus for surface modelling was to have a mathematical representation as faithful as possible to the designer’s mental perception of the surface; one is, however, hard pressed to find a similar motivation here. The formulation concepts used in Dutta’s work, which provide total freedom over the design of the material field in three-dimensional space, may prove cumbersome.

The design freedom is inherently limited by the designer's ability to process three-dimensional information. And although the human mind is well suited to visualize three-dimensional shapes and envelopes, the mental process of visualizing a volumetric distribution is more complex.

The approach of Jackson *et al.* (51), on the other hand, is based on a decomposition of the solid as a set of small cells. The material attributes are specified on a per cell basis, with the distribution function in each cell being defined by a blending function parametrized in barycentric co-ordinates. This approach is certainly sufficient to represent most material distributions, as long as the cells are small enough to accommodate the necessary design freedom. The decomposition used is arbitrary, but the type of cells is fixed (four vertices). The cell count is rather large; to correctly approximate the external envelope, a large number of cells is already necessary, and once a material distribution is chosen this number is increased even further.

Dutta and Jackson representations are somewhat diametrically opposed. Dutta provides a global view of the material distribution, as a combination of blending fields, while Jackson breaks down the solid into a large number of small elements whose composition is set arbitrarily. Dutta's representation develops on concepts with a proven track record, while Jackson's representation might be better suited to represent data issued mainly from an optimizer or a homogenizer. Other representations, relying on hatching patterns to emulate gradual changes between materials find their limitation in the complexity. There is no significant advantage of using a particular representation over another, as all meet specific needs and are aimed at different uses.

There is no clear consensus about what and how the matter should be represented, nor does there appear to be a description significantly better than another. Comparing the different representation methods lacks comparative grounds. Most of the prior research identified and presented in this chapter proposes an immediate solution to the problem of representing volumes of varying compositions with hardly any identification of the constraints, or a suitable specification of the problem at hand.

Furthermore, as multi-material manufacturing processes mature, so does the perception of the requirements and constraints of those processes. Already, it is observed that the control of the thickness and width of a *stroke* is difficult. Three-dimensional laser cladding proposes a feedback loop on the thickness of a layer, therefore bringing dynamics into the process. Similar concepts might redefine the process-planning task as a whole and put new constraints on slicing algorithms and on the underlying representation. In the case of Jackson's representation, the performance of an on-line slicer would be greatly dependent on the number of cells used. Similarly, Dutta's representation, which features multiple subdomains enclosed by complex surfaces, might be problematic for on-line treatment.

The work presented here introduces the essential features of current multi-material representation efforts. Most representations use a decomposition of the space, and define the blending of the material within each of these elements (Jackson, Dutta, voxel-based representations, octrees, and cell tree). The decomposition-free representations (Shepard method, geoscience modelling) lack the ability to represent the external boundary accurately.

The transfinite interpolation technique offers another possibility for the definition of a varying material composition between two boundaries. The distance functions created can be adjusted to be weighted with material components, thereby creating a varying composition field defined over the domains.

Looking at the future of CAD systems, it is uncertain whether designers will go to such lengths as to specify the exact blending function between two or more materials, though a global distribution might be defined as a preferential distribution (solid of revolution). Since the blending functions are dependent on the location considered in the solid and on the materials used, their actual definition might prove burdensome and difficult. It is very likely that a design automation tool will select an optimal and robust material field for a particular problem. It is also anticipated that the inherent approximations of this solution incurred at the manufacturing stage will have little effect on the overall performance of the part or that the representation will provide design rules to circumvent such peculiarities.

Even though existing representation schemes for heterogeneous objects provide means for representation of heterogeneous objects, they do not support the design of heterogeneous objects. The current methods for specifying material composition face a trade-off between the model coverage and operation convenience (55). They only provide a low-level description of geometry and material composition within the objects. They do not provide tools for designers to create and edit the heterogeneous object model.

The next section details one possible approach to multi-material representation. Chapters 7 and 8 give alternative approaches.

### **6.3 MMA-REP, A V-REPRESENTATION FOR MULTI-MATERIAL OBJECTS**

The following presentation takes a design approach to the representation problem, and comes up with a boundary representation (B-rep) based approach extended to a volumetric representation (V-rep).

#### **6.3.1 Design philosophy**

The objective in this research is to view the representation from a design perspective: the design of a representation to be used for multi-material solids. This requires the modelling constraints as well as the intended use for the representation to be carefully specified. To provide a uniform methodology in the design of such a representation, the approach is inspired from the one traditionally used for the design of products. The most influential sources of input in a design are as follows:

- Constraints and functionality
- Designer's knowledge of the problem
- Design prototyping
- Review of a design's fitness

Whether it is automated (optimizers) or manual (CAD packages), the design of the material distribution in a multi-material solid ought to account for the manufacturing characteristics during the decision process. Because of the variety of manufacturing processes available (additive/subtractive, powder/liquid medium, etc.), it is likely that each process will have different requirements and constraints. Ultimately, a given process will drive the optimization to a goal specific only to this process, while others may or may not reach a similar goal. It is expected that the optimizers will gain precise knowledge of

manufacturing process specifics independently from the representation, while it is hoped that only generalities will be gained by the representation itself.

Capturing material density information inside an object is a difficult problem, one that has not received sufficient attention thus far. The homogenization design method aims at obtaining a heterogeneous object model based on optimization, in which material composition is varied along with the geometry to achieve a desired functionality (56, 57). Homogenization has mostly considered material density variation in an element, and has only recently been investigated for multi-material variation. Such a method considers the effect of material composition variation upon function, and in general limits the role of the designer in the design process. It can be characterized as an implicit design method where designers do not have explicit control over material composition.

Although it is desirable to exploit as much information as possible from a particular multi-material apparatus or from a particular optimization process, both areas are highly active research topics, so only prevalent information is extracted in each case. Two types of constraints are defined:

- Accuracy-type properties, such as the size of the smallest features or grain size, to bind the spectrum of geometrical entities to be represented. There are a large number of constraints to be considered; the intent is not to draw an exhaustive list of those constraints, but rather to see their influence on the modelling.
- Practical considerations-type properties, such as the definition of a material space, eventually discretizing the spectrum of the data to be represented. Furthermore, the material compositions are coupled; design rules must be defined in order to account for feasibility considerations.

### **6.3.2 Designer's knowledge of the problem**

The body of research presented earlier serves as a basis for the resolution of the problem at hand. The basic formulation of the material space presented by Dutta is retained, along with some of the concepts outlined with the representations of colours and the accompanying colour spaces.



As a general rule, a representation featuring a minimal amount of computational overhead is preferred. Dynamic process planning such as that featured in three-dimensional laser cladding might prove to be more than a research tool, and the creation of such a feedback loop should not be precluded by a complex representation. The visual inspection of heterogeneous objects represented thus should also be permitted in a somewhat interactive manner, so that the evolution of the object can be seen. Visual inspection is seen as another form of manufacturing (or evaluation), except that it takes place on a computer monitor.

### **6.3.3 Material distribution data**

A newly designed part could conceivably be composed of multiple materials. The exact repartition of the material inside this solid can be assigned using optimization tools based on genetic algorithms (58, 59). Such optimization tools rely on a discretization of the geometric domain into smaller domains where the material composition can be changed. The difficulty in choosing a particular material distribution grows with the material space available for each subdomain.

From the optimization standpoint, every design variable input to an optimization problem affects a mechanical or thermal property (depending on the type of optimization problem), e.g. Young's modulus or thermal conductivity. The mapping between this property (mechanical or thermal) and material composition may be assumed to be linear. The conjecture made in such a case is that when 20 per cent of cobalt is mixed with 80 per cent of tungsten-carbide for instance, the resulting material properties (mechanical, thermal, and electrical) are 20 per cent those of cobalt added to 80 per cent those of tungsten-carbide. However, micro-scale finite element simulations (60) of those same materials, in agreement with experimental results, have shown considerable deviation from the linear law of mixtures. Other mixture rules (the power law especially) have shown better agreement, but research on proper characterization of heterogeneous components is still a hot topic.

MMA-rep represents volume fractions, not mechanical properties. The task of mapping a given material property to the volume fraction should be performed earlier on, preferably before the optimization.

The next section details the material space, and provides samples of two- and three-material spaces.

### 6.3.3.1 Material space

The formulation of the material space introduced by Dutta *et al.* is retained here:

$$V = \left\{ \mathbf{v} \in \mathbb{R}^n \mid \|\mathbf{v}\|_1 = \sum_{i=1}^n v_i = 1, v_i \geq 0 \right\}$$

Therefore for  $n$  materials,  $V$  is a subset of  $\mathbb{R}^n$ , the  $n$ -dimensional space formed by the set of all real numbers. For a given vector in  $V$ , the components  $v_i$  are positive real numbers summing to 1. The term nuance or shade is used interchangeably to designate a composition vector member of the material space  $V$ .

For a two-material arrangement, Fig. 6.12 shows the possible material spaces in increasing order of complexity, the red and green colour primaries are used for materials A and B respectively. There are roughly three types of approximations of this material space:

- Binary material space: there are only two possible nuances, either 100 per cent of material A or 100 per cent of material B.
- Discretized material space: there are a finite number of nuances in the composition (typical of some multi-material apparatus).
- Continuous material space: there are an infinite number of nuances.

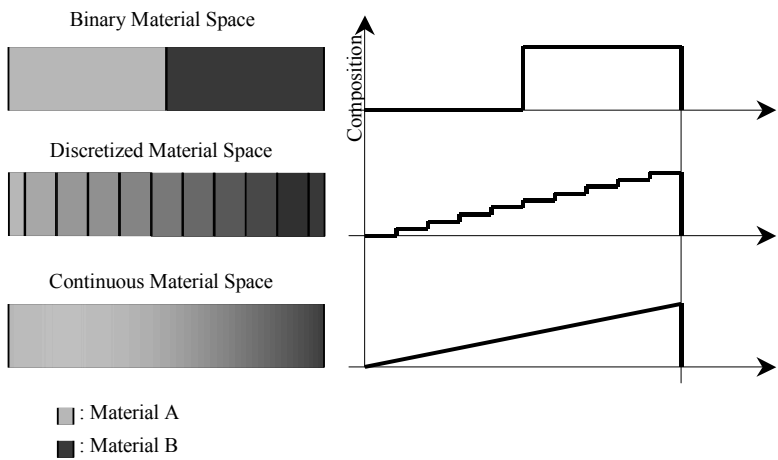
The binary material space is in fact a special case of the discretized material space, for only a pair of nuances.

Ideally, a discretized material space would reflect the capabilities of a given multi-material apparatus, therefore permitting the optimization to take place only in the space of feasible compositions. On both the discretized and continuous material spaces shown in Fig. 6.12, the nuances are uniformly arranged over  $V$ . This may not always be the case: some nuances may prove either unfeasible or impractical to manufacture.

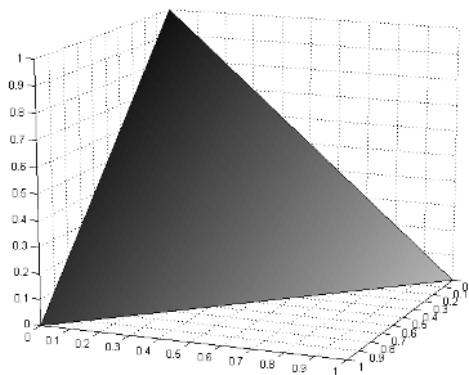
For a three-material arrangement,  $V$  is the portion of the plane restricted to the positive octant (e.g.  $x, y$ , and  $z \geq 0$ ) and with equation

$x + y + z = 1$ . Again, using the three red, green, and blue colour primaries for materials A, B, and C, respectively, the continuous space is represented in Fig. 6.13, with each axis representing one material.

The construction of a uniformly discretized material space is, however, more difficult for three materials. For any  $m = [a,b,c]$  desired material composition, there are three possible feasible compositions ‘near’  $m$ :  $\langle \cdot \rangle$  is the round function, which rounds a real number to the nearest integer (with the convention that  $\langle 0.5 \rangle = 1$ ).



**Fig. 6.12 Three possible material spaces for a two-material arrangement**



**Fig. 6.13 The continuous material space  $V$  for three materials (see also colour plate section)**

$$\langle \rangle: \begin{cases} u \rightarrow \langle u \rangle \\ \mathfrak{R} \rightarrow \mathfrak{S} \end{cases}$$

$$m_1 = \left[ 1 - \frac{\langle b \cdot n \rangle + \langle c \cdot n \rangle}{n}, \frac{\langle b \cdot n \rangle}{n}, \frac{\langle c \cdot n \rangle}{n} \right]$$

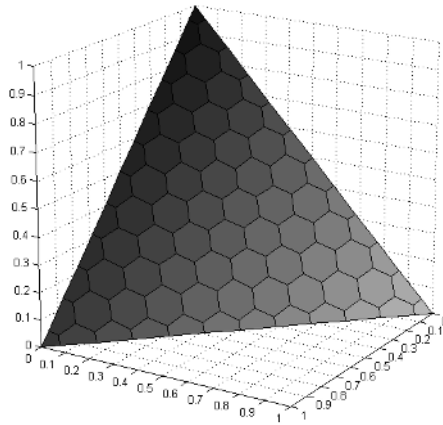
$$m_2 = \left[ \frac{\langle a \cdot n \rangle}{n}, 1 - \frac{\langle a \cdot n \rangle + \langle c \cdot n \rangle}{n}, \frac{\langle c \cdot n \rangle}{n} \right]$$

$$m_3 = \left[ \frac{\langle a \cdot n \rangle}{n}, \frac{\langle b \cdot n \rangle}{n}, 1 - \frac{\langle a \cdot n \rangle + \langle b \cdot n \rangle}{n} \right]$$

The triangle  $T$  formed by  $m_1$ ,  $m_2$ , and  $m_3$  is equilateral, since this is a uniform discretization of  $V$ . The three vertices  $m_i$ 's of  $T$  are all possible choices for the *nearest* feasible nuance from  $m$ . There is in general a unique choice for this feasible nuance which minimizes the Euclidean distance  $\|m - m_i\|_2$ ,  $i = 1, 2, 3$ , the exception being when  $m$  is exactly at equal distance from two or three vertices, e.g. when  $m$  is either on a median line or at the centre of the circumcircle of  $T$ , respectively.  $T$  can be divided into three partitions; based on the partition occupied by  $m$ , the nearest value is readily found. Repeating this approach for all feasible nuances in  $V$  generates Fig. 6.14. The feasible nuances are located at the centre of every hexagon.

In a non-uniform discretization of  $V$ , the feasible nuances might result from tests and experimentations on a multi-material apparatus; a desired nuance would be manufactured, its quality assessed (tested for strength, and thermal and electrical conductivity, etc.), and its composition measured. Based on the manufacturing success rate of this nuance, it would be considered for inclusion in the set of feasible nuances (61). In the case of a non-uniformly discretized material space, the approach presented earlier for the selection of the nearest feasible compositions can be generalized with the Voronoi diagram of all feasible nuances in  $V$ .

The Voronoi diagram is typically constructed for coplanar points, or in this case feasible nuances in  $V$ . For three materials,  $V$  is a portion of the plane defined by  $x + y + z = 1$ . For each feasible nuance  $m_f$  in  $V$ , there is a single boundary enclosing all the intermediate points lying



**Fig. 6.14 The discretized material space  $V$  for three materials**  
(see also colour plate section)

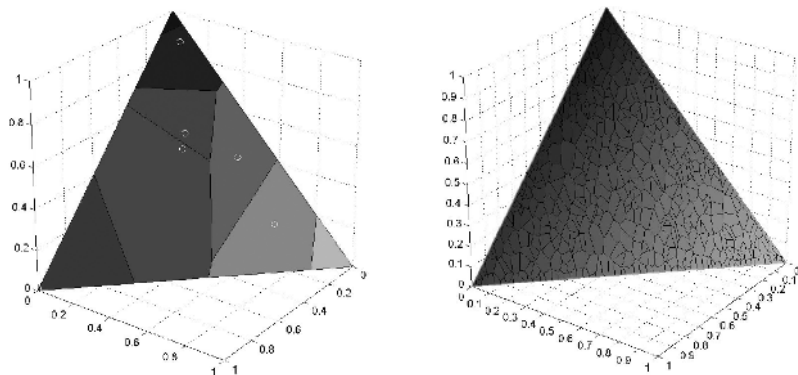
closer to  $m_f$  than to any other feasible nuance in the set. Such a boundary is called a *Voronoi polygon*, and the set of all Voronoi polygons for a given point set is called a *Voronoi diagram*. For an arbitrary distribution of respectively 8 and 400 feasible nuances in  $V$ , the corresponding discretized material spaces are shown in Fig. 6.15.

For a larger number of materials, the algorithm used for the creation of the Voronoi diagram must be extendable to higher dimensions, one such algorithm is the sweepline algorithm presented in reference (62).

A non-uniformly discretized parameter space has yet to be made available for research in multi-material optimization. In any case (uniformly or non-uniformly discretized), the number of nuances contained in  $V$  is a finite number known beforehand (versus an infinite number for the continuous material space).

### 6.3.3.2 Summary

For a given material distribution solution to an optimization problem, a shape is generally decomposed into subdomains. Each of these subdomains has a set material property assigned during optimization. The size of the solution space to a particular optimization problem precludes its parsimonious exploration; optimization therefore takes place over a rough approximation of this solution space.



**Fig. 6.15 Two non-uniformly discretized parameter spaces  $V$**   
(see also colour plate section)

The accuracy needs for a material distribution are twofold:

- The accuracy for the overall geometry or shape of the object represented.
- The accuracy of the material composition specified over each subdomain.

The approximations mentioned earlier (e.g. the fixed geometric decomposition as well as the discretization of the material space  $V$ ) make the accuracy needs of the material distribution subject to debate. Although it would be preferable to represent the material distribution with absolute accuracy, this may not be necessary. In effect, the definition of the material distribution should be taken in the large sense: instead of considering it strictly defined over subdomains, it should be guess-estimated at some locations (e.g. *inside* or *within* a given subdomain). How the domain is defined is the subject of the next section.

### 6.3.4 Interpolation space and interpolation kernel

Several further assumptions are needed to simplify the problem even more. These assumptions do not prevent common problems from being resolved but they do keep the interpolation problem simpler.

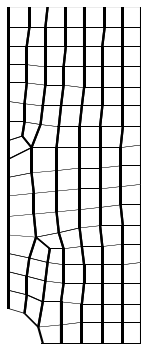
The data to be represented are in the form of a set of control composition points, each specified by co-ordinates in the geometric space (either two- or three-dimensional). A known order exists in the arrangement of the control composition points. From this order, a set of two- or three-dimensional entities mapping the space enclosed by

the boundary can be extracted. A mesh such as that used in FEA is suitable to provide this order. This arrangement of points, or mesh, is homeomorphic to a possibly degenerate mapped mesh.

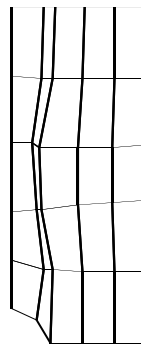
This latter condition ought to be further detailed. A mapped mesh is a type of mesh used in FEA (Fig. 6.16). These meshes constrain the number of nodes throughout the domain to remain constant across opposed boundaries. This does limit the type of geometry that can be interpolated, as described later. Interpolating a mapped mesh, where the control composition points are specified at each node, can be done with a scheme presented later. Other types of material distribution can however be approximated in this representation, using some of the methods presented earlier as an aid to conversion.

The mapped mesh may exhibit degenerate elements resulting from the replications of nodes to match the node count in a given direction.

There is no requirement to have the nodes equidistant. The other requirements are to have the entire mesh described by a network of adjacency relations fitting on a two- or three-dimensional grid. Each of the nodes located inside the mesh have four and only four neighbouring nodes (possibly duplicates). Nodes located on the edges have three and only three neighbouring nodes (possibly duplicates). Finally, corner nodes have two and only two neighbouring nodes; there can be only  $2^d$  corner nodes ( $d = 2$  and  $d = 3$  for two- and three-dimensions respectively.)



Free Mesh



Mapped Mesh

**Fig. 6.16 Free and mapped mesh examples**

Thus the actual data to be represented are as follows:

- An enclosing boundary described by way of a mathematical formulation.
- A set of points inside or on the enclosing boundary, specifying the material composition as an  $n$ -dimensional vector described over a grid.

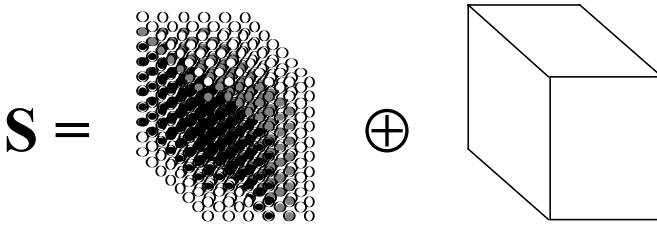
Find a representation scheme for  $S = \{(M, \mathbf{v}) \in \mathbb{R}^3 \times V\}$ , where

$V$  is the material space defined earlier, i.e.

$$V = \left\{ \mathbf{v} \in \mathbb{R}^n \mid \|\mathbf{v}\|_1 = \sum_{i=1}^n v_i = 1, v_i \geq 0 \right\}$$

$M$  is the set of points located both inside and on the outer surface(s) of the solid

Figure 6.17 represents, from a conceptual standpoint, the data to be interpolated. The following sections will outline the necessary requirements for an interpolation scheme for these types of data.



**Fig. 6.17 Data to be interpolated**

#### 6.3.4.1 Interpolation in $n + 3$ dimensions

The data to be interpolated are arranged over a regular or irregular grid, and are as follows:

At every triplet  $(x_i, y_j, z_k)$  a composition  $\mathbf{v}_{i,j,k} = ([v_{i,j,k}]_0, [v_{i,j,k}]_1, \dots, [v_{i,j,k}]_d)$  is defined such that



$$\| \mathbf{v}_{i,j,k} \|_1 = 1 \quad \text{and} \quad \begin{cases} 0 \leq i < N_x \\ 0 \leq j < N_y, & N_x, N_y, \text{ and } N_z \geq 1 \\ 0 \leq k < N_z \end{cases}$$

The interpolation must be performed by way of three parameters (two for two-dimensional interpolation). The interpolation is performed concurrently on both the geometrical data and the control compositions, yet the choice of the blending functions used in the geometrical data is independent from those used in the interpolation of the material composition.

### 6.3.5 Representation summary

A theoretical representation for multi-material solids has been presented and the design issues of this representation have been outlined, including their impact on the data accuracy and completeness. The data are represented by way of an interpolation in  $n + 3$  dimensions, where  $n$  is the number of materials, and is parametrized by three parameters. The selection of the parameter space and some of its shortcomings have been exposed; the benefits embedded in this choice are proven in the following sections.

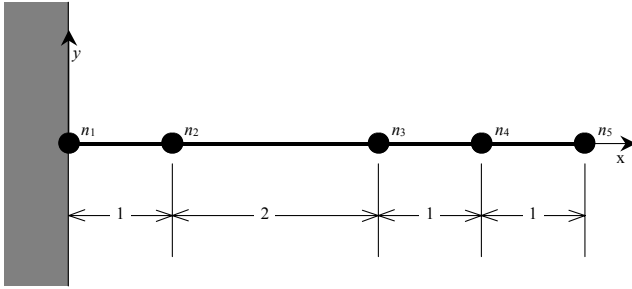
The interpolation is performed concurrently on both the geometry and the material by way of an independent, but similar, interpolation method. The geometric interpolation does not preclude the use of more elaborate geometric interpolation techniques, and the material interpolation is constrained to operate in the continuous material space  $V$ . A linear interpolation is performed on the geometry, although a better interpolation quality may be desired. The linear interpolation produces line segments (in two dimensions) or polygons (in three dimensions) and was selected because of the *modus operandi* of current RP technologies.

To better understand the features of this representation for multi-material solids, termed MMA-rep, several applications for one-, two-, and three-dimensional solids are considered. The intent is to show that the constraints outlined above are respected and to explore how they are put to use in practical situations.

## 6.4 APPLICATIONS

### 6.4.1 One-dimensional solids

A one-dimensional solid is dependent on a single parameter only, though it can occupy a two- or three-dimensional space. A typical one-dimensional solid is a constant cross-section beam such as those found in strength-of-materials textbooks, or a solid of revolution. A sample beam is shown in Fig. 6.18.



**Fig. 6.18** Sample one-dimensional geometry

At each of the nodes  $n_i$  defining the geometry, the corresponding abscissa is defined:  $x_1 = 0$ ,  $x_2 = 1$ ,  $x_3 = 3$ ,  $x_4 = 4$ , and  $x_5 = 5$ , respectively. A parametric representation of the abscissa is given by

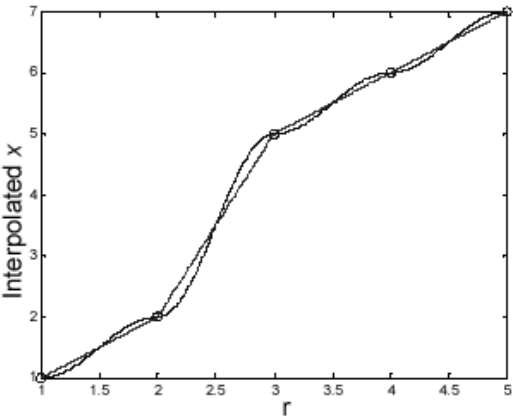
$$x(r) = \sum_{i=1}^5 X_i(r) \cdot x_i \quad \text{with } 1 \leq r \leq 5$$

noting that each geometrical interpolating function  $X_i(r)$  is defined with respect to a single interpolating kernel,  $X_i(r) = k_g(r - i)$  (the choice of a uniform parameter space implies  $r_i = i$ ). Using this representation and expanding the previous sum translates into

$$\begin{aligned} x(r) = & 0 \cdot k_g(r - 1) + 1 \cdot k_g(r - 2) + 3 \cdot k_g(r - 3) + 4 \cdot k_g(r - 4) \\ & + 5 \cdot k_g(r - 5) \end{aligned}$$

The choice of the geometric blending kernel  $k_g(r)$  affects how the geometric interpolation is performed; Fig. 6.19 shows two types of interpolations, performed with different blending kernels. However, to respect the original constraints set on the geometric interpolation (e.g.

to perform as well as a polygonal approximation), the geometric interpolation must be linear, thus restricting the choice of the geometric kernel to  $k_{\text{linear}}$ .



**Fig. 6.19 Sample one-dimensional interpolation using two different kernels**

At each of the nodes  $n_i$ , an arbitrary composition is set. For a two-material geometry, a set of compositions is selected arbitrarily as shown below.

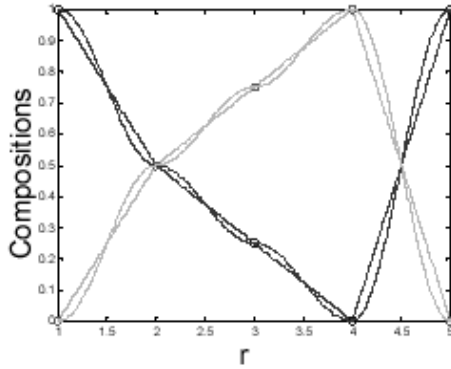
Node	Material 1	Material 2	$\mathbf{v}$
			(Comp. vector)
$n_1$	1	0	$\mathbf{v}_1=(1,0)$
$n_2$	0.5	0.5	$\mathbf{v}_2=(0.5,0.5)$
$n_3$	0.25	0.75	$\mathbf{v}_3=(0.25,0.75)$
$n_4$	0	1	$\mathbf{v}_4=(0,1)$
$n_5$	1	0	$\mathbf{v}_5=(1,0)$

The parametric material composition interpolation is as follows

$$\mathbf{v}(r) = \sum_{i=1}^5 R_i(r) \cdot \mathbf{v}_i \quad \text{with} \quad 1 \leq r \leq 5$$

A material blending kernel is used, and the choice of a uniform parameter space translates into  $R_i(r) = k_m(r - i)$ .

Using the red and green colours for the first and second materials respectively, the material composition interpolation can be plotted as a set of two functions (one for each material components). Several blending kernels meet the continuity conditions required. A linear and second-order kernel are used to represent the sets of material interpolation functions in Fig. 6.20. These functions are complementary, since the composition components always sum to 1.



**Fig. 6.20 One-dimensional interpolation for two materials**

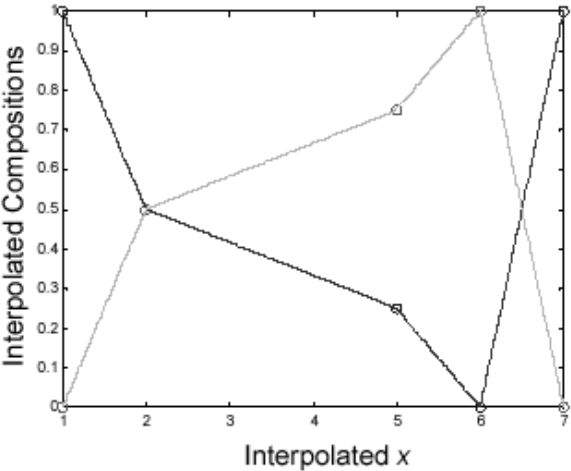
#### **6.4.1.1 *Material composition interpolation versus geometric interpolation***

The interpolation presented earlier can now be plotted as a parametric plot of one variable, with the beam abscissa  $x(r)$  along the horizontal axis and the material composition components,  $v_1(r)$  and  $v_2(r)$  along the vertical axis, using the red and green colour, respectively, to differentiate both curves. A first set of curves, using the linear blending kernel  $k_{\text{linear}}$  for both the geometric and material interpolation is presented in Fig. 6.21.

Since linear blending kernels were used in both cases, the parametric plot remains in essence linear. Performing the material interpolation in a non-linear manner shows however a different trend for the curves, as show in Fig. 6.22. The behaviour at the node points is explained by elementary studies on the derivability of parametric curves.

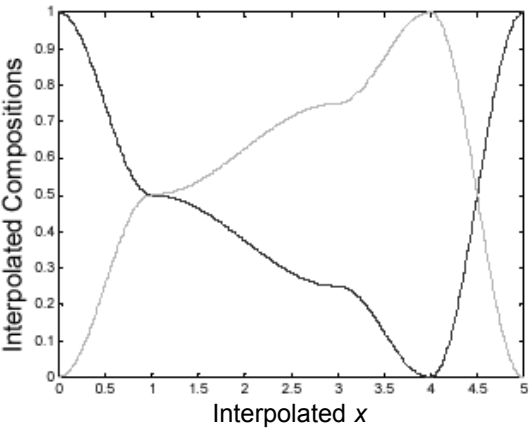
#### **6.4.1.2 *Visualizing an interpolated material distribution***

For solids composed of fewer than three materials, the variation of the material versus the geometry can be captured directly on the geometry,



**Fig. 6.21 Parametric plot of material versus geometric interpolation**

using the red, green, and blue colour primaries to emulate blending of the material components. The intermediate material nuances produced during interpolation thereby generate shades of colours resembling a varying material composition. A sample representation of the beam presented earlier is shown in Fig. 6.23, the red colour represents the first material, while the green colour represents the second material.



**Fig. 6.22 Material versus composition interpolation using different blending kernels**



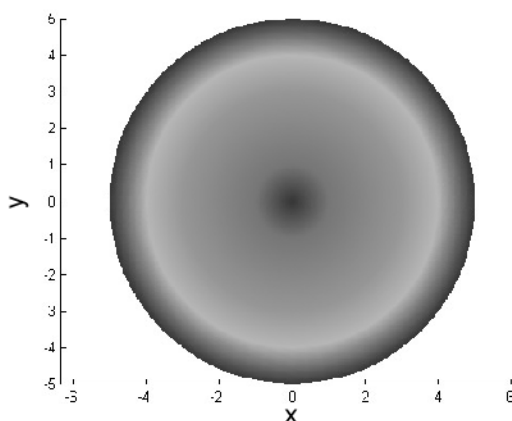
**Fig. 6.23** Material composition depicted by colour shades

This approach is especially appealing for material distributions that would be otherwise difficult to visualize. For instance, applying the previous material distribution to a solid of revolution generates a solid with a radial material distribution. Considering that the variation of the material components over the circular domain would require a family of three-dimensional plots (each material component, having to be plotted versus the  $x$  and  $y$  co-ordinates, generates a three-dimensional plot), the use of colour primaries to convey the distribution is valuable and necessitates only a single plot (Fig. 6.24).

#### **6.4.1.3 Representing material discontinuities**

A solid may sometimes exhibit an abrupt material transition, termed binary transition, varying from 100 per cent of say material A, to 100 per cent of say another material B. This binary transition can be accomplished by replicating the geometrical components of the node where the transition occurs, while applying the changed composition at the second 'replicated' node.

Considering again the previous beam as an example, a binary transition is implemented at node  $n_2$ : a node  $n_2'$  is inserted between



**Fig. 6.24** The radial material distribution

node  $n_2$  and  $n_3$ . The parameter space is modified accordingly:  $1 \leq r \leq 6$ . The abscissa  $x_2'$  of node  $n_2'$  is the same as that of node  $n_2$ :

$$x(r) = 0 \cdot k_g(r-1) + 1 \cdot k_g(r-2) + 1 \cdot k_g(r-3) + 3 \cdot k_g(r-4) \\ + 4 \cdot k_g(r-5) + 5 \cdot k_g(r-6)$$

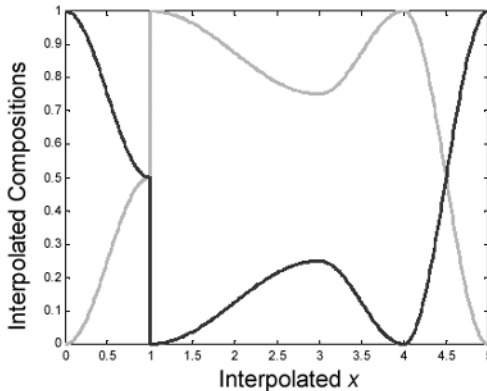
The material composition however is different at nodes  $n_2'$  and  $n_2$ . Choosing an arbitrary value for  $v_2'$  such as (0,1) yields the following material distribution:

$$v(r) = v_1 \cdot k_m(r-1) + v_2 \cdot k_m(r-2) + v_2' \cdot k_m(r-3) + v_3 \cdot k_m(r-4) \\ + v_4 \cdot k_m(r-5) + v_5 \cdot k_m(r-6)$$

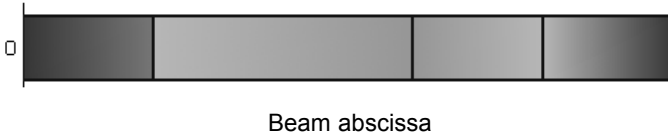
Again, representing both  $v_1(x)$  and  $v_2(x)$  on a single plot yields the plot shown in Fig. 6.25, clearly showing the binary transition occurring at node  $n_2$ . As before, the material map can be plotted using the red and green colour primaries to emulate the varying material distributions (Fig. 6.26).

#### 6.4.1.4 Application to one-dimensional heterogeneous flywheel

A flywheel is a mechanical component that stores kinetic energy. The mass distribution of a flywheel around its axis of revolution impacts its capacity to store kinetic energy. A high-energy storage is typically attained whenever the mass is distributed as far as possible from the axis of revolution (e.g. on the circumference of the flywheel). Huang and Fadel (59) proposed a novel approach to the design of a flywheel:



**Fig. 6.25** A sharp material transition



**Fig. 6.26 A sharp material distribution represented using colours**

essentially, since this is a solid of revolution, the problem is reduced to the design of its cross-section. The geometry along with the composition in a cross-section is designed by blending two materials with opposed properties, namely high strength/low weight and low strength/high weight.

The design is therefore treated as an optimization problem and the objectives that ought to be met are:

- Maximize kinetic energy stored
- Minimize maximal von-Mises equivalent stress
- The design is limited by dimensional constraints (maximum height, internal and external radius)

The composition is represented by way of the volume fractions of material A with respect to material B [ $v_a = V_a / (V_a + V_b)$ ]. The geometry and the composition are represented using parametric equations featuring Bezier blending functions:

$$z(u) = \sum_{i=0}^n z_i \cdot B_i^n(u)$$

$$v_a(u) = \sum_{i=0}^n v_a^i \cdot B_i^n(u)$$

$$r(u) = \sum_{i=0}^n r_i \cdot B_i^n(u)$$

$$B_i^n(u) = \frac{n!}{i! \cdot (n-i)!} \cdot u^i \cdot (1-u)^{n-i}, \quad 0 \leq u \leq 1$$

The  $r_i$  are set and represent the location of the control points along the radius for the geometry and the composition,  $z_i$  and  $v_i$ , respectively. Both the  $z_i$  and  $v_i$  represent the design variables. The optimizer



computes the optimum values for all  $2(n + 1)$  design variables, and attempts to meet the objectives presented earlier.

The materials selected are tungsten-carbide (heavy/low strength) and cobalt (light/high strength), 22 control points are selected, and a sample result is presented in Fig. 6.27.

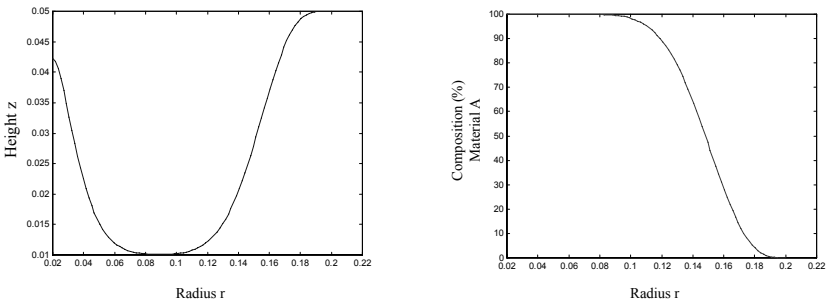
The Bezier blending functions are not interpolating functions: they only pass through the endpoints, not the intermediate (control) points. To use the results presented with MMA-rep, the curve constituting the cross-section must be sampled to generate node points. However, if the computed control points are inserted in lieu of the node points, and if the geometric blending functions are chosen to be Bezier blending functions, the sampling step is bypassed and the geometrical interpolation is performed directly using Bezier blending functions. Similarly, since the composition was computed for two materials, the volume fraction of material B is obtained from the volume fraction of A using the following relation:

$$v_a(u) + v_b(u) = \sum_{i=0}^n v_a^i \cdot B_i^n(u) + \sum_{i=0}^n v_b^i \cdot B_i^n(u) = \sum_{i=0}^n (v_a^i + v_b^i) \cdot B_i^n(u) = 1$$

The previous equation must hold for any value of  $u$ , thus

$$v_a^i + v_b^i = 1$$

The composition at each point is defined by the volume fractions of material A and material B. The previous relation permits computation

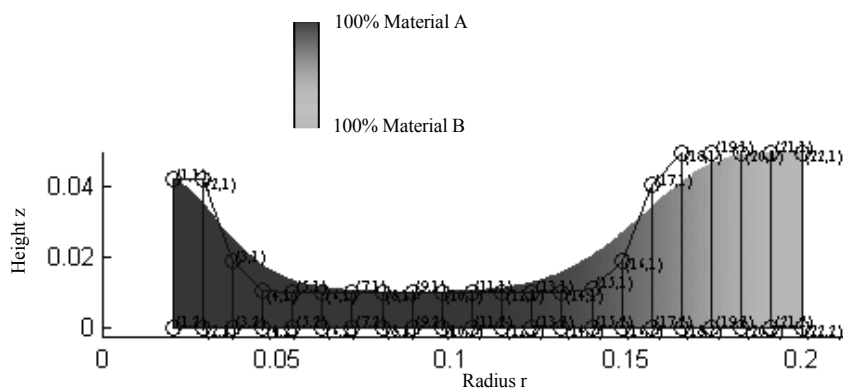


**Fig. 6.27 Optimal geometry and composition for a heterogeneous flywheel**

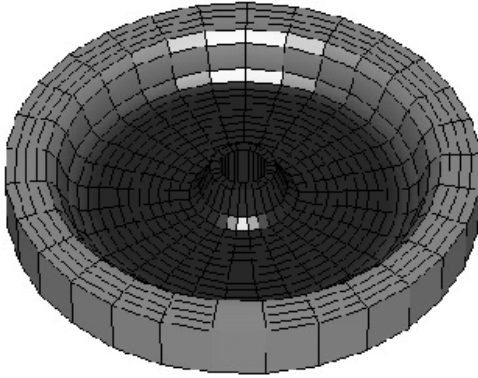
of the composition at any point based on the composition at every control point. Again, the control composition at each node can be computed by sampling the control composition curves, and by using the sampled points as node points to perform interpolation. Like the geometrical interpolation, the control composition can be used directly in lieu of the node points, and the Bezier blending function can be used in lieu of the material blending function. The cross-section generated according to this approach is presented in Fig. 6.28. The numbering of the nodes shows their arrangement in the parameter space.

In that particular case, and because the optimization was performed directly using Bezier control points, it is suitable to use Bezier blending functions for both the geometry and the material composition. However, the Bezier blending functions are not suitable as geometric or material blending kernels; therefore, unless the data input for MMA-rep comes in the form of Bezier control points, this approach cannot be systematically used. For more general cases, both the profile curve and the material composition must be sampled, and the sampled points can be used as node points in the MMA-rep (approximations will however result).

This discussion is in fact reminiscent of the practice of interpolation versus approximation of curves as discussed in solid modelling and computer graphics textbooks (63–65). Additionally, the issue of having the entire net of nodes contributing to the value of every point is also subject to debate; this is in fact embedded in the choice of the blending



**Fig. 6.28 MMA-rep of heterogeneous flywheel using Bezier interpolation**



**Fig. 6.29 Three-dimensional view of the heterogeneous flywheel  
(see also colour plate section)**

kernels. For scattered-type data, the choice of a blending kernel with local support may prevent error propagation during interpolation (66).

A three-dimensional view of the solid obtained by revolving the cross-section presented in Fig. 6.28 is shown in Fig. 6.29.

#### **6.4.1.5 Summary**

The essential features of this representation for heterogeneous solids have been introduced, including simple means to visualize the representation. The selection of the material and geometrical blending kernel must be carefully considered in order to ease sampling and ensure fidelity of the representation. The inclusion of sharp transitions allows representation of heterogeneous solids featuring binary transitions, gradual transitions, or solids exhibiting both types of transitions.

The interfacing with optimization-type data has been investigated. The approach used by Huang and Fadel (59) for the heterogeneous flywheel optimization, which concurrently optimizes the shape and material distribution by taking the geometric and control polygon as design parameters, might allow a closer integration between representation and optimization. Huang chose to use Bezier blending functions for both the material and the geometry, which might be sub-optimal as Bezier blending functions lack local support – a typical trade-off for this class of function.

The choice of these blending functions ensures membership in  $V$ , the material space. A curve represented by way of Bezier blending functions features the convex-hull property, which bounds the latter curve to the convex hull of the control polygon. Having the composition profile contained in the convex-hull of the control polygon ensures composition membership in  $V$ , as the constraints are shifted from the composition profile to the nodes of the (composition) control polygon.

Furthermore, at every sampled point within the inner and outer radius, the computation of the height or composition considers the entire set of design variables (e.g. the control polygon). Investigating a class of material blending kernels with only local support (as opposed to global support for the Bezier blending functions) could help provide a class of functions simplifying the optimization strategy, reinforcing the importance of the careful selection of kernel blending functions.

### 6.4.2 Two-dimensional solids

In this section, the basic principles behind this representation will be used to represent a more general class of solids: two-dimensional solids, where both geometry and material distribution are described entirely with two parameters. Additionally, simple Boolean operations on solids represented using MMA-rep are presented.

#### 6.4.2.1 Heterogeneous flywheel

A hypothetical flywheel composed of three materials is created, featuring *spokes* of varying compositions. The core of the flywheel is mounted on a hypothetical shaft and features a binary material transition. The geometry is that of a disc with an inner diameter  $r_i$  ( $r_i = 2$  in) and external diameter  $r_e$  ( $r_e = 6$  in), while the material composition is defined by way of a function in polar co-ordinates:

$$\mathbf{v}(r, \theta) = [f_1(r, \theta), f_2(r, \theta), f_3(r, \theta)]$$

The composition is in the material space, so

$$\mathbf{v} \in V \Rightarrow f_1(r, \theta) + f_2(r, \theta) + f_3(r, \theta) = 1$$

The construction of the material composition functions follows. To feature the number of spokes, the functions are made periodic in their

angular argument  $\theta$ , with a period of  $360^\circ/5 = 72^\circ$ . To select a particular material for the circumference, the Heaviside  $H(t)$  function is useful:

$$H(t) = \begin{cases} 1, & t > 0 \\ 0, & t < 0 \end{cases}$$

The composition is adjusted to feature the desired transition zones, using the  $\min()$  and  $\max()$  functions. The thickness of the inner core is set to be  $t_i$ , while the thickness of the material on the circumference is set to  $t_e$ .

$$f_1(r, \theta) = m_1(r, \theta') - mf_1(r, \theta')$$

$$f_2(r, \theta) = 1 - m_1(r, \theta')$$

$$f_3(r, \theta) = mf_1(r, \theta')$$

With the auxiliary functions  $m_1(r, \theta')$  and  $mf_1(r, \theta')$  defined as

$$m_1(r, \theta') = \max[mf_1(r, \theta'), mf_2(r, \theta')]$$

$$\begin{cases} mf_1(r, \theta') = H((r_i + t_i) - r) \\ mf_2(r, \theta') = \max[\text{Bound}(1 - (\theta'/15)^2), \text{Bound}(r - (r_e - t_e))] \end{cases}$$

with

$$\theta' = \theta \left[ \frac{2\pi}{5} \right] \quad \text{and} \quad \text{Bound}(u) = \begin{cases} 1, & u > 0 \\ t, & 0 \leq u \leq 1 \\ 0, & u < 0 \end{cases}$$

All of these auxiliary functions handle the change in the composition as  $(r, \theta)$  parse  $[r_i, r_e] \times [0, 2\pi]$ . This composition function can now be sampled, and the nodes created. A rendering of this cross-section is presented in Fig. 6.30. Again, to handle the binary transition located near the shaft, the same principle (replication of the geometry, while modifying the composition) is used.

#### 6.4.2.2 Constructive solid geometry

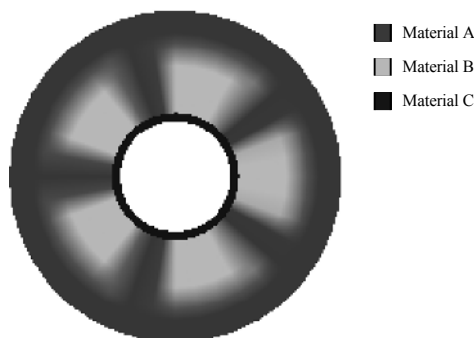
Now that the basic features of MMA-rep have been introduced, it is relevant to study the arrangement of nodes at the parameter space

level. For a two-dimensional solid, the nodes are arranged on a grid in the parameter space. New nodes can be inserted in the parameter space, requiring relocation of the original nodes, along with other adjustments. This next example shows how this can be realized for a simple geometry.

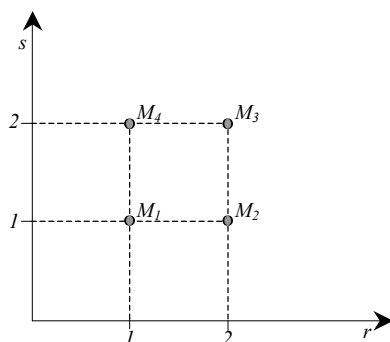
A square element is composed of four nodes  $M_1, M_2, M_3, M_4$ . The four nodes are arranged as follows in the parameter space (Fig. 6.31).

Associating the  $(r,s)$  pair corresponding to each of the  $M_i$  nodes can be done by locating the  $M_i$ 's in a matrix:

$$C = \begin{bmatrix} M_1 & M_2 \\ M_4 & M_3 \end{bmatrix}$$



**Fig. 6.30 Heterogeneous flywheel composed of three materials**  
(see also colour plate section)



**Fig. 6.31 Node arrangement in the parameter space for a square element**

Thus, the row and column associated to every node located in  $C$  specifies the  $(r,s)$  pair associated with each node. With  $\mathbf{K}_r$  and  $\mathbf{K}_s$  representing the interpolating vector in the  $r$  and  $s$  directions, respectively, the computation of an interpolated point at a location  $(r,s)$  in the parameter space is realized with the computation of the point  $M(r,s)$ :

$$\begin{aligned} M(r,s) &= {}^t\mathbf{K}_r \cdot \begin{bmatrix} M_1 & M_2 \\ M_4 & M_3 \end{bmatrix} \cdot \mathbf{K}_s \\ &= \begin{bmatrix} k_1(r) & k_2(r) \end{bmatrix} \cdot \begin{bmatrix} M_1 & M_2 \\ M_4 & M_3 \end{bmatrix} \cdot \begin{bmatrix} k_1(s) \\ k_2(s) \end{bmatrix} \end{aligned}$$

with

$$\mathbf{K}_r = \begin{bmatrix} k_1(r) \\ k_2(r) \end{bmatrix} \quad \text{and} \quad \mathbf{K}_s = \begin{bmatrix} k_1(s) \\ k_2(s) \end{bmatrix}$$

Since every geometric or material blending function is defined by way of a geometric or material blending kernel, respectively, the computation of a geometrical component is as follows:

$$x(r,s) = \begin{bmatrix} k_g(r-1) & k_g(r-2) \end{bmatrix} \cdot \begin{bmatrix} x_1 & x_2 \\ x_4 & x_3 \end{bmatrix} \cdot \begin{bmatrix} k_g(s-1) \\ k_g(s-2) \end{bmatrix}$$

Evidently for  $y(r,s)$ , the  $y_i$  co-ordinates are switched for the  $x_i$  co-ordinates:

$$y(r,s) = \begin{bmatrix} k_g(r-1) & k_g(r-2) \end{bmatrix} \cdot \begin{bmatrix} y_1 & y_2 \\ y_4 & y_3 \end{bmatrix} \cdot \begin{bmatrix} k_g(s-1) \\ k_g(s-2) \end{bmatrix}$$

For the  $k$ th material component:

$$v_k(r,s) = \begin{bmatrix} k_m(r-1) & k_m(r-2) \end{bmatrix} \cdot \begin{bmatrix} v_{1,k} & v_{2,k} \\ v_{4,k} & v_{3,k} \end{bmatrix} \cdot \begin{bmatrix} k_m(s-1) \\ k_m(s-2) \end{bmatrix}$$

For a simple 3-by-3 square element constituted of 100 per cent of material A, the nodes are  $M_1 = (0,0,1,0)$ ,  $M_2 = (3,0,1,0)$ ,  $M_3 = (3,3,1,0)$ ,

and  $M_4 = (0,3,1,0)$ . The first two components represent the  $x$  and  $y$  coordinates, while the other two represent the material composition.

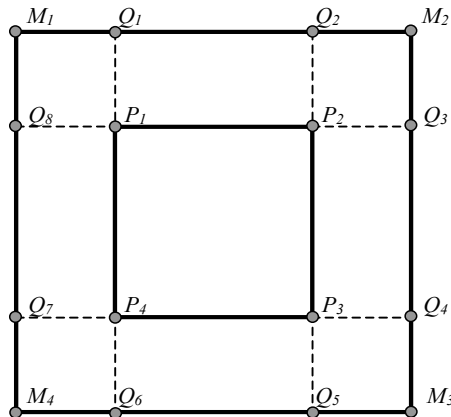
The inclusion of a 1-by-1 constant-composition square element in the original element results in modifications to  $C$ :

$$C = \begin{bmatrix} M_1 & Q_1 & Q_2 & M_2 \\ Q_8 & P_1 & P_2 & Q_3 \\ Q_7 & P_4 & P_3 & Q_4 \\ M_4 & Q_6 & Q_5 & M_3 \end{bmatrix}$$

Where  $P_1 = (1,1,0,1)$ ,  $P_2 = (2,0,0,1)$ ,  $P_3 = (2,2,0,1)$ , and  $P_4 = (0,2,0,1)$ . The  $Q_i$ s ‘fill in’ the newly created nodes. Because of the conservation of the original geometry through this inclusion, the nodes  $Q_1$  and  $Q_2$  must be located on the segment  $(M_1, M_2)$ . As long as the order of the points on the segment is  $(M_1, Q_1, Q_2, M_2)$  (corresponding to parameter pairs  $(1,1)$ ,  $(2,1)$ ,  $(3,1)$ , and  $(4,1)$ , respectively), any location on that segment is suitable for  $Q_1$  and  $Q_2$ . The geometries under consideration lend themselves well to the following definition for the missing points (see Fig. 6.32).

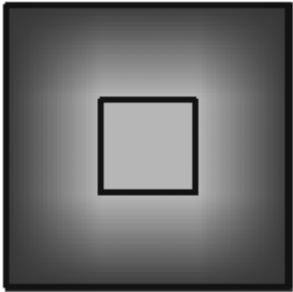
The insertion of the 1-by-1 square results in the geometry presented in Fig. 6.33.

The material interpolation performed is valid at all points in this heterogeneous square. The result differs from the traditional



**Fig. 6.32 Definition of additional node points**



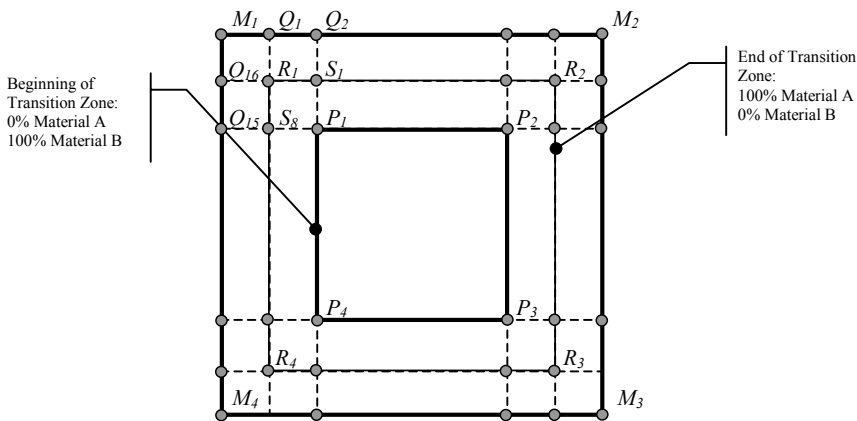


**Fig. 6.33 Result of the insertion**

CSG/Boolean operator ‘minus’ in that it does not generate a binary material transition at the interface of the two solids, but rather permits development of a zone where the material is blending from 100 per cent of material A (red) to 100 per cent of material B (green).

This transition zone between the two materials can be constrained to accommodate design specifications. Control of the transition zone can be obtained by inserting a constant-composition square (100 per cent of material A) delimiting the end of the transition zone from material B to material A, as shown in Fig. 6.34.

Inserting the square defined by  $R_1$ ,  $R_2$ ,  $R_3$ , and  $R_4$  changes  $C$  into



**Fig. 6.34 Bounds on the transition zones**

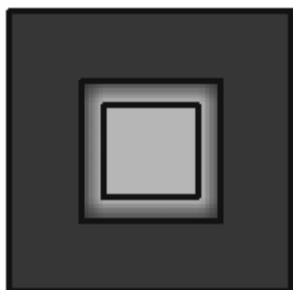
$$C = \begin{bmatrix} M_1 & Q_1 & Q_2 & Q_3 & Q_4 & M_2 \\ Q_{16} & R_1 & S_1 & S_2 & R_2 & Q_5 \\ Q_{15} & S_8 & P_1 & P_2 & S_3 & Q_6 \\ Q_{14} & S_7 & P_4 & P_3 & S_4 & Q_7 \\ Q_{13} & R_4 & S_6 & S_5 & R_3 & Q_8 \\ M_4 & Q_{12} & Q_{11} & Q_{10} & Q_9 & M_3 \end{bmatrix}$$

The  $Q_i$  and  $R_i$  values are intermediate points that must be created to accommodate the inclusion of the two elements. Again, the geometries used lend themselves well to the selection of those points as shown in Fig. 6.34 (to prevent clutter, only a fraction of those points are labelled). The result, showing the now constrained transition zone, is shown in Fig. 6.35.

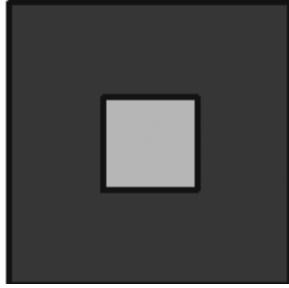
Reverting to traditional CSG/Boolean operations, e.g. generating binary material transitions, is then straightforward: the boundaries of both the start and end of the transition zone are made identical. This results in the introduction of replicated nodes, following accordingly from the earlier remarks concerning the implementation of binary transitions. The result of such a binary operation is presented in Fig. 6.36.

### 6.4.2.3 Summary

The representation of two-dimensional heterogeneous solids featuring complex material distributions can be achieved using MMA-rep in a variety of ways. Whether the material distribution is a two-variable



**Fig. 6.35** Constraining the transition zone



**Fig. 6.36 Reverting to traditional Boolean operator**

function defined over a geometry, resulting from optimization-type data, or from conventional CSG/binary operations, simple considerations are sufficient to represent these solids using MMA-rep.

The low complexity associated with this representation translates into a significant storage cost. When solid inclusion is performed, additional nodes must be created within the original geometry to accommodate the new inclusion. Every element inclusion translates into several new nodes to be created and computed; if the  $C$  matrix is originally  $n_{\text{row}}$  rows by  $n_{\text{col}}$  columns, inserting a single node results in the creation and computation of  $n_{\text{row}} + n_{\text{col}}$  new nodes, while the  $C$  matrix is enlarged to be  $n_{\text{row}} + 1$  rows by  $n_{\text{col}} + 1$  columns. For simple cases, the constrained inclusion of a square element in an  $n_{\text{row}} \times n_{\text{col}}$   $C$  matrix, requires approximately a total of  $4(n_{\text{row}} + n_{\text{col}} + 4)$  nodes to be created and computed, while the  $C$  matrix is enlarged to contain  $n_{\text{row}} + 4$  rows by  $n_{\text{col}} + 4$  columns.

This supplemental storage cost is acceptable and is in fact likely to be amortized when more complex geometries are represented, as the nodes inserted and not directly participating in the implementation of the constraints (nodes  $Q_i$  and  $S_i$  on Fig. 6.34) can be reused to implement other constraints.

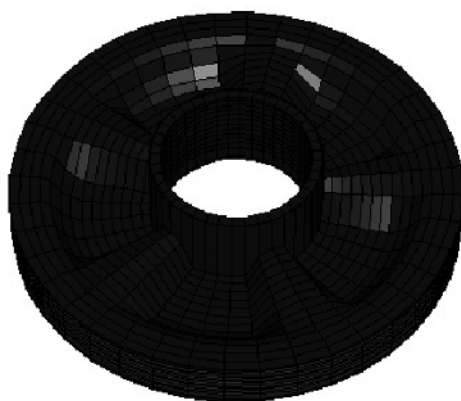
### 6.4.3 Three-dimensional solid

The MMA-rep is now used to represent a three-dimensional extension of the flywheel introduced earlier. Again, the material distribution function is created in closed form and is building on similar considerations to those outlined earlier. The overall geometry is

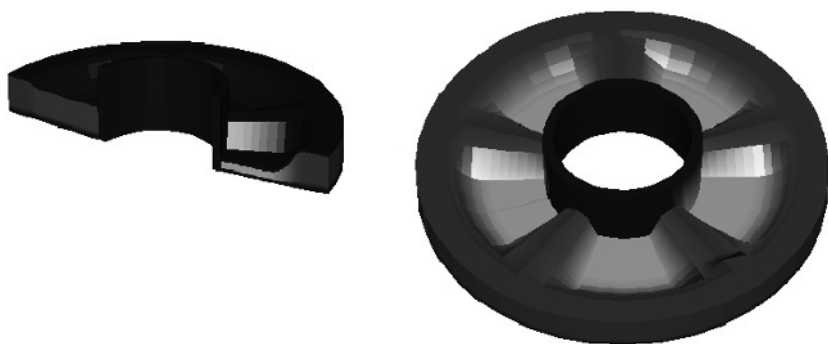
reminiscent of a cylindrical solid. The solid is encased in the third (blue) material, while the varying composition remains confined to the inner part of the flywheel (Fig. 6.37).

The innards of this flywheel can be viewed with more ease using cross-sections or by removing layers of nodes (Fig. 6.38). The blending of the materials between nodes is shown only at relevant locations.

As heterogeneous solids become commonplace in mechanical CAD, so will the need to visualize the actual material distribution within the solid. The visualization techniques used in volume rendering [generation of arbitrary cross-sections, removal of specific materials,



**Fig. 6.37 Three-dimensional heterogeneous flywheel**  
(see also colour plate section)



**Fig. 6.38 Cross-sections and details of three-dimensional flywheel**  
(see also colour plate section)

use of transparency, see reference (67)] might provide such means, though work remains to be done to attain interactive visualization capabilities.

The three-dimensional flywheel represented here featured a total of  $41 \times 25 \times 15$  nodes. Having an indexed representation of the parameter space pointing to a given node in a node database required the storage of 15375 additional integers, and  $6 \times 15375 = 92250$  floating point numbers. The overall storage cost of representation for this geometry, for 4-byte encoding of integers and 8-byte encoding of floating point numbers, required a total of  $15375 \times 4 + 92250 \times 8 \approx 800$  kb.

#### **6.4.4 Summary of MMA-rep features**

There is a wide array of solids that can be represented using MMA-rep. The constructs commonly used in CAD to build solids, such as the CSG/binary operators, can be implemented to operate directly on MMA-rep without breaking the solid modelling thought process.

The selection of the material blending kernel must be done with practical considerations in mind. The sampling of the composition along, say, a toolpath, must be done with ease and accuracy. Therefore the material blending kernel must be considered along with the geometrical blending kernel, and issues of derivability, typically easing sampling, must be addressed as well.

The issue of storage requirement is deemed somewhat secondary: a typical low-end CAD workstation often features a minimum of 256 Mb of random access memory (RAM), thus allowing complex geometries to be represented.

It has also been shown how MMA-rep can be modified to operate on other types of control points, again, without breaking major constructs. The interfacing of MMA-rep with optimization results permits unification of the representation of the solution to heterogeneous solids, and furthermore, to expand the span of problems that can be treated as such.

## 6.5 CONCLUSION

This chapter attempts to relay to the reader the difficulty of the representation of the material composition inside an object made of multiple materials. The solutions presented, as well as those discussed in the next chapters, are workable solutions that will undoubtedly evolve and converge to some robust and flexible representation format. We attempted to make the reader aware of the various issues to consider in trying to represent heterogeneous objects. It is our hope that the standard will emerge from this presentation.

## REFERENCES

- 1 Poynton, C. (2000) Frequently-asked questions about colour. <http://www.inforamp.net/~poynton/ColorFAQ.html>
- 2 Golland, P. and Bruckstein, A. M. (1996) Why R.G.B.? Or how to design color displays for Martians. *Graphical Models and Image Processing*, **58**, 405–412.
- 3 Norman, R. B. (1990) Color models. In *Electronic Color: 'The Art of Color Applied to Graphic Computing*, Ch. 3, p. 183, (Van Nostrand Reinhold, New York).
- 4 CIE (1986) Colourimetry, Publication 15.2, Compagnie International de L'Eclairage, Vienna, Austria.
- 5 Fraser, B. (1998) The full gamut: getting the most from Photoshop 5.0's new RGB working spaces. *Adobe Magazine*.
- 6 Emsley, H. H. (1952) *Visual Optics*, 5th edition (Hatton Press, London).
- 7 Burrough, P. A. and Frank, A. U. (1996) *Geographic Objects with Indeterminate Boundaries* (Taylor & Francis, London).
- 8 Fisher, P. (1996) Boolean and fuzzy region. In *Geographic Objects with Indeterminate Boundaries* (Eds P. A. Burrough and A. U. Frank), pp. 87–94 (Taylor & Francis, London).
- 9 Usery, E. L. (1996) A conceptual framework and fuzzy set implementation for geographic features. In *Geographic Objects with Indeterminate Boundaries* (Eds P. A. Burrough and A. U. Frank), pp. 71–85 (Taylor & Francis, London).
- 10 Sabella, P. (1988) A rendering algorithm for visualizing 3D scalar fields. *Computer Graphics*, **22**, 51–59.
- 11 Banvard, R. A. and Ackerman, M. J. (1996) In Proceedings of The Visible Human Project Conference, William H. Natcher Conference Center, Bethesda, Maryland (National Institutes of

- Health, Bethesda, Maryland).
- 12 Möller, T., Machiraju, R., Mueller, K., and Yagel, R. (1997) A comparison of normal estimation schemes. Presented at the 8th IEEE Visualization, Phoenix, Arizona (IEEE Computer Society).
  - 13 Freund, J. and Sloan, K. (1997) Accelerated volume rendering using homogeneous region encoding. Presented at the 8th IEEE Visualization, Phoenix, Arizona (IEEE Computer Society).
  - 14 Mueller, K. and Yagel, R. (1996) Fast perspective volume rendering with splatting by utilizing a ray-driven approach. In *Proceedings of Visualization '96*, San Francisco, California, pp. 65–72 (IEEE Computer Society).
  - 15 Shareef, N. and Yagel, R. (1995) Rapid previewing via volume-based solid modelling. Presented at the Third Symposium on *Solid Modelling and Applications*, Salt Lake City.
  - 16 Chandru, V., Manohar, S., and Prakash, C. E. (1995) Voxel-based modelling for layered manufacturing. *IEEE Computer Graphics and Applications*, **15**(6), 42–47.
  - 17 Samet, H. (1989) *The Design and Analysis of Spatial Data Structures* (Addison-Wesley, Reading, Massachusetts).
  - 18 Ayala, D., Brunet, P., Juan, R., and Navazo, I. (1985) Object representation by means of non-minimal division quadtrees and octrees. *ACM Transactions on Graphics*, **4**, 41–59.
  - 19 Meagher, D. (1982) Geometric modelling using octree encoding. *Computer Graphics and Image Processing*, **19**, 129–147.
  - 20 BSP Tree Frequently Asked Questions (FAQ) (2001). <http://reality.sgi.com/bspfaq>
  - 21 Eberly, D. H. (2001) *3D Game Engine Design: A Practical Approach to Real Time Computer Graphics* (Morgan Kaufmann, Los Altos, California).
  - 22 Günther, O. (1988) *Efficient Structures for Geometric Data Management* (Springer-Verlag, New York).
  - 23 O'Rourke, J. (2000) The comp.graphics.algorithm FAQ. [ftp://rtfm.mit.edu/pub/usenetby/hierarchy/comp/graphics/algorithms/comp.graphics.algorithms\\_Frequently\\_Asked\\_Questions](ftp://rtfm.mit.edu/pub/usenetby/hierarchy/comp/graphics/algorithms/comp.graphics.algorithms_Frequently_Asked_Questions)
  - 24 Shepard, D. (1968) A two dimensional interpolation function for irregularly spaced points. Presented at the 23rd ACM National Conference, New York (Association for Computing Machinery).
  - 25 Renka, R. J. (1988) Multivariate interpolation of large sets of

- scattered data. *ACM Transactions on Mathematical Software*, **14**, 139–148.
- 26 Melenk, J. M. and Babuska, I. (1996) The partition of unity finite element method: basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, **139**, 289–314.
- 27 Belytschko, T., Krongauz, K., Organ, D., Fleming, M., and Krysl, P. (1996) Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, **139**, 3–47.
- 28 Bloomenthal, J. (1997) *Introduction to Implicit Surfaces* (Morgan Kaufmann Publishers, Los Altos, California).
- 29 Entz, M. T., Storti, D. W., and Ganter, M. A. (1998) Implicit methods for geometry creation. *International Journal of Computational Geometry and Applications*, **8**, 509–536.
- 30 Rvachev, V. L., Sheiko, T. I., Shaipro, V., and Tsukanov, I. (2000) Transfinite interpolation over implicitly defined sets, Technical Report SAL2000-1, Spatial Automation Laboratory, University of Wisconsin-Madison.
- 31 Shapiro, V. (1991) Theory of R-functions and applications: a primer, Department of Computer Science, Cornell University, Ithaca, New York.
- 32 Stanton, E. L., Crain, L. M., and Neu, T. F. (1997) A parametric cubic modeling system for general solids of composite material. *International Journal for Numerical Methods in Engineering*, **11**, 653–670.
- 33 Requicha, A. A. G. and Voelcker, H. B. (1983) Solid modelling: current status and research directions. *IEEE Computer Graphics Applications*, **3**(7), 25–37.
- 34 Hoffmann, C. M. (1989) *Geometric and Solid Modeling: An Introduction* (Morgan Kaufmann Publishers, Los Altos, California).
- 35 Pratt, M. (1988) Synthesis of an optimal approach to form feature modeling. In Proceedings of the ASME Conference on *Computers in Engineering*, San Francisco, August.
- 36 Rossignac, J. and O'Connor, M. (1989) SGC: a dimension-independent model for pointsets with internal structures and incomplete boundaries. In *Geometric Modeling for Product Engineering* (Eds M. Wosny, J. Turner, and K. Preiss), Proceedings of the IFIP Workshop on *CAD/CAM*, pp. 145–180 (North-Holland).
- 37 Kumar, V. and Dutta, D. (1998) An approach to modeling and



- representation of heterogeneous objects. *Journal of Mechanical Design*, **120**, 659–667.
- 38 Middleditch, A. E., Reade, C. M. P., and Gomes, A. J. (1999) Set-combinations of the mixed dimensional cellular objects of the Djinn API. *Computer-Aided Design*, **31**, 683–694.
  - 39 Jackson, T. R., Liu, H., Patrikalakis, N. M., Sachs, E. M., and Cima, M. J. (1999) Modeling and designing functionally graded material components for fabrication with local composition control. *Materials and Design*, **20**, 63–75.
  - 40 Wu, Z., Soon, S. H., and Lin, F. (1999) NURBS-based volume modeling. International Workshop on Volume Graphics, pp. 321–330.
  - 41 Park, S.-M., Crawford, R. H., and Beaman, J. J. (2000) Functionally gradient material representation by volumetric multi-texturing for solid freeform fabrication. In Solid Freeform Fabrication Symposium, Austin, Texas, 7–9 August.
  - 42 Kumar, A.V. and Wood, A. (1999) Representation and design of heterogeneous components. Presented at the Solid Freeform Fabrication Symposium, Austin, Texas.
  - 43 Kumar, V. and Dutta, D. (1997) Solid model creation for materially graded objects. Presented at the Symposium on *Solid Freeform Fabrication*. Austin, Texas.
  - 44 Corney J. (1997) *3D Modelling with the ACIS Kernel and Toolkit* (John Wiley, New York).
  - 45 Birch-Holt, J., Koizumi, M., Hirai, T., and Munir, Z. A. (1993) *Functionally Gradient Materials*, **34**.
  - 46 Qian, X. and Dutta, D. (1998) Features in the layered manufacturing of heterogeneous objects. In Symposium of Solid Freeform Fabrication, The University of Texas at Austin, August, pp. 689–696.
  - 47 STEP (ISO 10303-1): Overview and fundamental principles (International Standards Organization).
  - 48 NIST, The STEP Project (2000). <http://www.nist.gov/sc4/www/stepdocs.htm>
  - 49 Patil, L., Dutta, D., Bhatt, A. D., Jurens, K., Lyons, K., Pratt, M. J., and Sriram, R. D. (2000) Representation of heterogeneous objects in ISO 10303 (STEP). Presented at ASME Manufacturing Engineering Division, Orlando, Florida.
  - 50 Kumar, V. (1999) Solid modeling and algorithms for heterogeneous objects. PhD thesis, Mechanical Engineering, University of Michigan.

- 51 Jackson, T. R., Patrikalakis, N. M., Sachs, E. M., and Cima, M. J. (1998) Modeling and designing components with locally controlled composition. Presented at the Solid Freeform Fabrication Symposium, Austin, Texas.
- 52 Jackson, T. R. (2000) Analysis of functionally graded material object representation methods. PhD thesis, Massachusetts Institute of Technology, June.
- 53 Sachs, E., Haggerty, J., Cima, M., and Williams, P. (1995) Three Dimensional Printing Techniques, US: 5,204,055.
- 54 Brisson, E. (1993) Representing geometric structures in d dimensions: topology and order. *Discrete Computational Geometry*, **9**, 387–426.
- 55 Qian, X. (2001) Feature methodologies for heterogeneous object realization. PhD dissertation, Mechanical Engineering, University of Michigan, April.
- 56 Bendsøe, M. P. and Kikuchi, N. (1988) Generating optimal topologies in structural design using a homogenisation method. *Computer Methods in Applied Mechanics and Engineering*, **71**, 197–224.
- 57 Cherkaev, A. and Kohn, R. (Eds)(1997) *Topics in Mathematical Modelling of Composite Materials* (Birkhäuser Verlag, Basel).
- 58 König, O. (1999) Application of genetic algorithms in the design of multimaterial structures manufactured in rapid prototyping. Diploma thesis, Institute for Construction and Design, ETH Zurich, Switzerland; Department of Mechanical Engineering, Clemson University, USA.
- 59 Huang, J. and Fadel, G. M. (2000) Heterogeneous flywheel modeling and optimization. *Journal of Materials and Design*, **21**, 111–125.
- 60 Grujicic, M., Cao, G., and Fadel, G. M. (2001) Effective materials properties: determination and application in mechanical design and optimization. *Journal of Materials, Design and Applications*, **215**(L4), 225–234.
- 61 Love, J., Keicher, D., and Morvan, S. (2000) Optomec Design Company (personal communication).
- 62 Fortune, S. (1987) A sweep-line algorithm for Voronoi diagrams. *Algorithmica*, **2**, 153–174.
- 63 Farin, G. (1990) *Curves and Surfaces for Computer Aided Geometric Design* (Academic Press, San Diego, California).
- 64 Mortenson, M. E. (1985) *Geometric Modelling*, 1st edition (John Wiley, New York).

- 65 Foley, J. D., van Dam, A. and Feiner, S. (1990) *Computer Graphics: Principles and Practice* (Addison-Wesley).
- 66 Watson, D. F. (1992) *CONTOURING: a Guide to the Analysis and Display of Spatial Data* (Pergamon Press).
- 67 Lichtenbelt, B., Crane, R., and Naqvi, S. (1998) *Introduction to Volume Rendering* (Prentice Hall).

# Chapter 7

## Feature methodologies for heterogeneous object realization

**X Qian<sup>1</sup> and D Dutta<sup>2</sup>**

<sup>1</sup>xpqian@engin.umich.edu

<sup>2</sup>dutta@engin.umich.edu

Department of Mechanical Engineering, University of Michigan,  
Ann Arbor, MI 48109-2125

### 7.1 INTRODUCTION

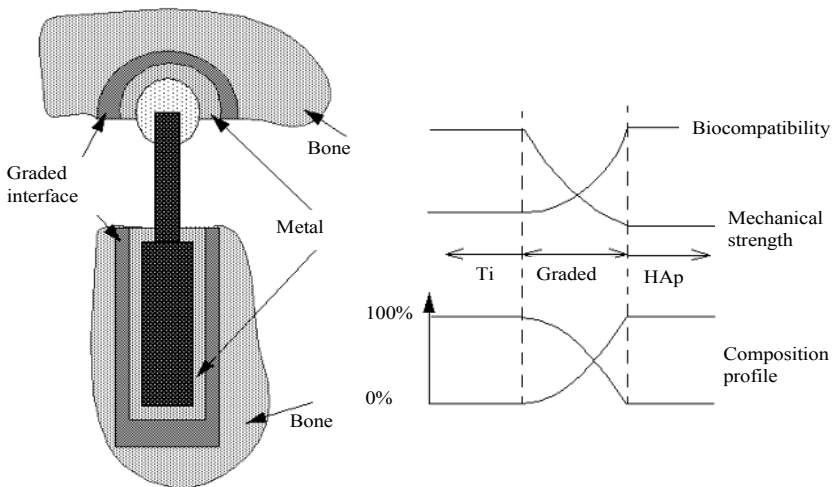
Heterogeneous objects are composed of different constituent materials. They are sometimes known as functionally gradient materials (FGM). They have the ability to exhibit continuously varying composition and/or microstructure, thus producing a gradation in their properties. Such material gradation can be tailored to achieve multiple functionalities and to satisfy conflicting design requirements. These properties in general cannot be achieved by using one single material.

For example, a prosthesis using the graded interface in the orthopaedic implant is shown in Fig. 7.1. Conventional methods of fixing an artificial bone and joining the prosthesis to the bone include total close contact of the prosthesis to the bone. However, this causes pain to the patient during weight bearing because there is micromotion of the prosthesis within the bone, and subsequently the prosthesis may even loosen in the bone. A more effective method for adhering a prosthesis to the bone is to coat it with a porous metal because new bone grows into the pores after the implantation. A graded layer of hydroxyapatite

(HAp) is coated on the porous metal. It bonds to the bone physicochemically, thereby increasing the adhesion strength and the rate of binding to the bone. Therefore, porous metal with an HAp coating remedies the drawbacks of cementless prosthesis. It prevents pain to the patient caused by micromotion while walking, or loosening of a prosthesis fixed without the bone cement.

Figure 7.1 is a schematic structure of such an FGM interface. This FGM region is composed of porous titanium plus hydroxyapatite (HAp). Ti has good mechanical toughness and HAp has good biocompatibility. Simple combination of Ti and HAp would cause bio-incompatibility and weakened strength due to their material property differences. Such material property differences are resolved by using a mixture of Ti and HAp with varying proportions. The sharp interface between the Ti and HAp is eliminated by using a graded zone of Ti/HAp. The bending strength of the resulting material is similar to human bone.

As evidenced in this example, many applications based on the concept of functionally gradient materials can be developed to exploit the multiple desirable material properties. In order to have mass applications of heterogeneous objects, systematic methodologies are needed.



**Fig. 7.1 Schematic structure of an FGM interface within a prosthesis**

Nonetheless, the existing methods for the design and fabrication of heterogeneous objects tend to be experimental and *ad hoc*.

The state-of-the-art research on heterogeneous object modelling has been primarily focusing on representation schemes for heterogeneous objects. Currently, there is only limited means available to obtain heterogeneous object models. There are not yet any design tools available for designer to design heterogeneous objects. Existing methods are mostly implicit methods, in which designers do not have explicit control over geometry and material composition variation.

In addition to the need for a design methodology, fabrication methodology for heterogeneous objects is also needed for mass applications of heterogeneous objects. The existing fabrication methods are grossly inadequate for the fabrication of heterogeneous objects when objects are of intricate geometry and complex material composition. They are inept for handling the fabrication of heterogeneous objects where material variations are three dimensional through the object space. A recent technique, layered manufacturing, can fabricate objects with three-dimensional material variation. However, there is still a lack of efficient fabrication methods for these processes.

Therefore, to enable mass applications of heterogeneous objects, effective and systematic methodologies are needed for heterogeneous object realization. In this research, we propose the use of features to facilitate the design and fabrication processes for heterogeneous objects.

Features were initially proposed to automate the link between design and numerical control (NC) path generation (1). Since then, feature techniques have been widely and successfully used in CAD/CAM systems. Feature-based design expedites the design process, and feature recognition facilitates the fabrication process planning. A feature-based product model also simplifies the assembly, inspection planning, and other downstream applications (2).

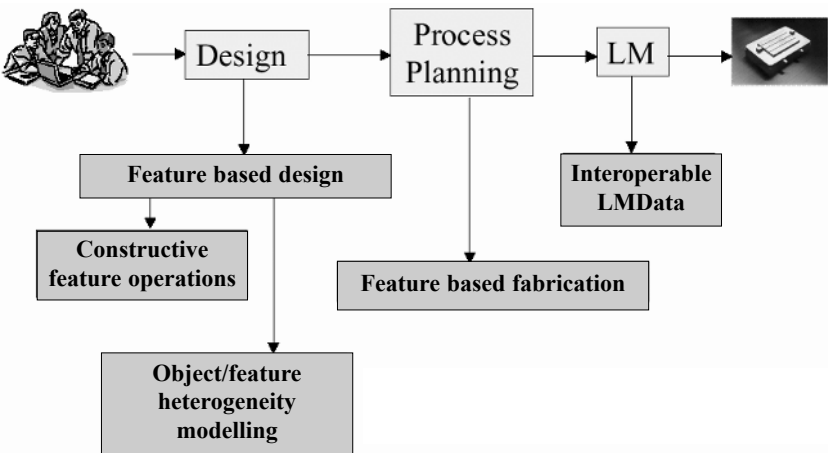
In this chapter, we present feature-based design and fabrication methodologies for heterogeneous object realization. Accomplishing these research objectives provides the following set of enabling tools to facilitate the heterogeneous object realization (Fig. 7.2). In heterogeneous object realization, three stages of activities are needed:

design, process planning, and fabrication. This research provides methodologies for both design and process planning tasks, while the interoperable layered manufacturing data (LMData) (3) address the data exchange issues in layered manufacturing.

The remainder of this chapter is organized as follows. In Section 7.2, we present a review of the existing research relating to the design and fabrication of heterogeneous objects. In Section 7.3, a general methodology – feature-based design for heterogeneous objects – is presented. Sections 4 and 5, on the other hand, present two enabling component techniques for feature-based design: direct face neighbourhood alteration for constructive feature operations and physics-based modelling for feature/object heterogeneity modelling. Section 6 presents the feature-based fabrication methodology for the layered manufacturing of heterogeneous objects. Finally, Section 7 summarizes this paper and identifies future work.

## 7.2. LITERATURE REVIEW

Many representation schemes have been developed to represent solids. Manifold solids and R-sets were first proposed to represent solid models (4,5). A radial-edge data structure is another data structure for modelling non-manifold solids (6). For conventional feature modelling, the usage of non-manifold structures was first proposed by



**Fig. 7.2 Computer enabling tools for heterogeneous object realization**

Pratt (7). A selected geometric complex (SGC) is a non-regularized non-homogeneous point set represented through enumeration as a union of mutually disjoint connected open cells (8). Constructive non-regularized geometry (CNRG) was also proposed to support dimensionally non-homogeneous, non-closed point sets with internal structures (9). Middleditch *et al.* (10) present mathematics and formal specification for the mixed dimensional cellular geometric modelling. The cellular model provides a geometric basis for heterogeneous object modelling.

Current research on heterogeneous objects has led to many representation schemes for heterogeneous object modelling. Kumar and Dutta (11) proposed R-m sets be used for representing heterogeneous objects. Jackson *et al.* (12,13) proposed another modelling approach based on subdividing the solid model into subregions and associating the analytical composition blending function with each region. Some other modelling and representation schemes, such as utilizing voxel models, implicit functions, and texturing, have also been proposed (14–16).

Even though existing representation schemes for heterogeneous objects provide means to represent heterogeneous objects, they do not support the design of heterogeneous objects. The current methods for specifying material composition face a trade-off between the model coverage and operational convenience (17). They only provide a low-level description of geometry and material composition within the objects. They do not provide tools for designers to create and edit the heterogeneous object model.

Currently, there are only limited means available to obtain heterogeneous object models. Reverse engineering converts existing objects into computer representations. Such an approach can be utilized to obtain a heterogeneous object model. However, these techniques typically represent heterogeneous objects only in the discretized format: three-dimensional voxel (18,19). Research has also been conducted to convert the three-dimensional images to three-dimensional geometric objects. Nonetheless, the focus has been on recreating the outer geometry. The issues on material composition modelling have not been directly addressed. Capturing material density information inside is a difficult problem, one that has not received sufficient attention thus far.



The homogenization design method is another method of obtaining a heterogeneous object model based on optimization, in which material composition is varied along with the geometry to achieve the desired functionality (20,21). These methods consider the effect of material composition variation upon function.

In general, these methods limit the role of the designer in the design process. They can be characterized as implicit design methods where designers do not have explicit control over material composition.

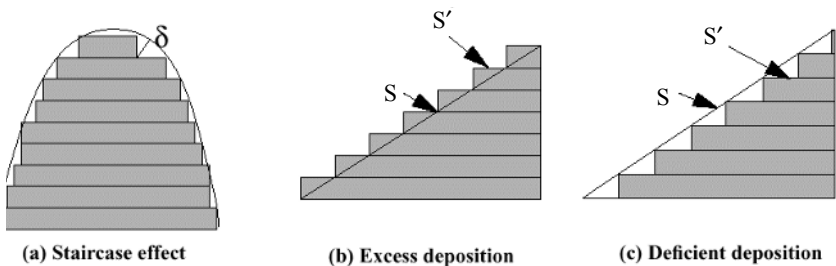
The early methods for the fabrication of heterogeneous objects include powder metallurgy, physical and chemical vapour deposition, plasma spraying, self-propagating high temperature synthesis (SHS), and galvanofarming. Recently, several fabrication methods have been developed that are capable of manufacturing heterogeneous objects in which the material variations are three dimensional. These new fabrication methods can be broadly referred to under the term 'layered manufacturing' (LM). They fabricate parts by depositing materials layer-by-layer under computer control. A host of LM technologies are currently available commercially. A non-exhaustive list includes: stereolithography (SLA) from 3D Systems, selective laser sintering (SLS) from DTM Corp., fused deposition modelling (FDM) from Stratasys Corp., solid ground curing (SGC) from Cubital, and laminated object manufacturing (LOM) from Helisys. In addition, several LM processes are under development at various universities, such as Carnegie Mellon, Stanford, MIT, University of Dayton, University of Michigan, and the University of Texas. Refer to reference (22) for details of these LM processes.

Layer-wise fabrication in LM leads to the staircase effect for slant surfaces, as shown in Fig. 7.3a. (Here  $\delta$  controls the cusp height, the maximum distance between the nominal part boundary and the boundary of the part produced by LM.) Depending on the intended application of the LM part one would, in general, employ excess deposition or deficient deposition (23). Figures 7.3b and c show the two deposition situations, in which  $S$  is the part boundary and  $S'$  is the boundary of the part produced by LM.

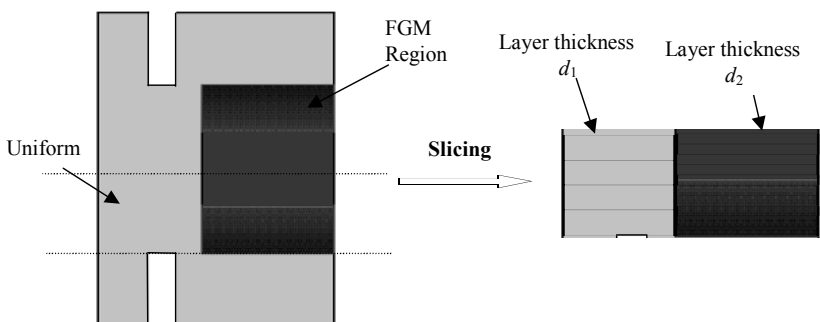
This layer-wise stack in LM has one inherent drawback – the staircase effect. To have better surface quality, a thinner layer thickness is desired. On the other hand, the thinner the layers are, the more layers

it takes to build the part and the more build time it takes. To overcome the conflicting requirements associated with high surface quality and low building time, adaptive slicing was developed (23–25). The idea was to decrease slice thickness in high-curvature (in terms of normal value in slice direction) regions and to increase slice thickness in relatively flat regions. Sabourin (26) proposed a variant whereby adaptive high-precision exterior and high-speed interior can be achieved by layered manufacturing. We term the fabrication inefficiency due to the geometry curvature the *geometry curvature effect*.

In the meantime, material variation in the objects further complicates the layer thickness computation. Existing methods are inefficient in this since they fabricate heterogeneous objects with the minimum layer thickness of all the building blocks (27). For example, the example part in Fig. 7.4 is composed of two different types of building blocks, one of a uniform material, the second made of functionally gradient material. Even though the geometric curvature of the two building blocks is the same and would allow for the same layer thickness,



**Fig. 7.3 Staircase effect and different deposition situations**



**Fig. 7.4 Material variation complicates the layer thickness computation**

however, due to material variation in the FGM region, it requires thinner layers than the counterpart in the uniform regions, i.e.  $d_2 < d_1$ . Existing methods would use  $d_2$  as the layer thickness for the entire object, which leads to extra deposition time over the region of uniform material. We term such fabrication inefficiency due to the material gradation in different building blocks the *material gradation effect*.

Both the geometric curvature effect and the material gradation effect are resolved in this research by localizing the effects within each feature layer.

### 7.3 FEATURE-BASED DESIGN FOR HETEROGENEOUS OBJECTS

In this section, we examine the relationships between form features and material features in heterogeneous objects. We synthesize the form features and material features and then propose the constructive feature-based design for heterogeneous objects.

#### 7.3.1 Features

Feature techniques have traditionally only focused on the geometry, i.e. form features. Because of the nature of material variation in heterogeneous objects, we shall examine features not only in terms of the geometry but also in terms of the material composition.

In order to mathematically represent the features, we first define the notation. A part,  $P(G,M)$ , is defined as a product space, where  $G$  is the geometry and  $M$  is the material space.

##### 7.3.1.1 Form feature

A form feature is a specific geometric shape which carries engineering significance, such as a hole and a slot. A form feature can be either a volume feature or a surface feature. In this paper, we focus on volume features.

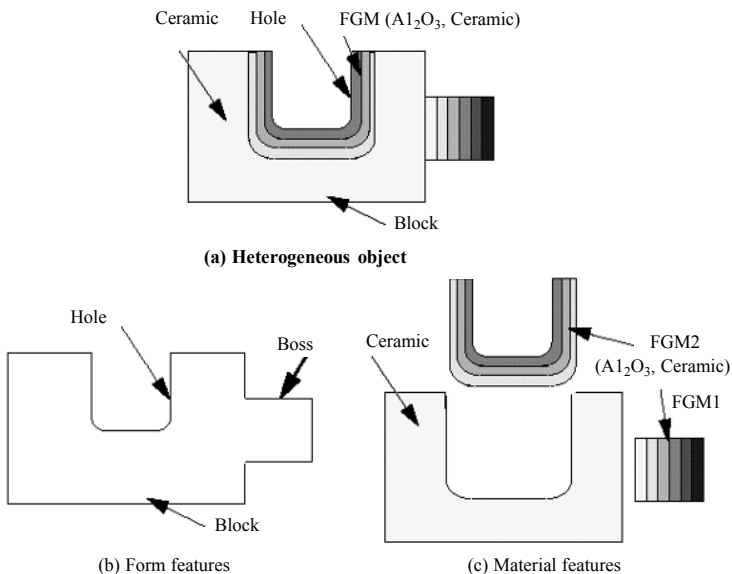
As with homogeneous objects, a form feature in a heterogeneous object is a specific shape within a part regardless of the material composition variation. It should be noted that, in order to distinguish form features from material features, there are two necessary conditions to the definitions of form features. First, *the shape of the*

*volume must correspond to some specific engineering meaning. For example, form features such as a hole or a groove have specific geometric shapes. Second, such a shape should contribute to the formation of the exterior boundary of the final part geometry. That is to say, during the part creation process, the evolving part geometries should be different before and after the introduction of the form features. We note the part geometry as  $G_i$  before the form feature  $FF_{i+1}$  is introduced to the part. We have the necessary condition for form features:  $FF_{i+1} - G_i \neq \emptyset$ .*

For example, in Fig. 7.5, the heterogeneous object has three form features: a block, a hole, and a boss. They each represent a particular geometric shape. If we disregard the material variation in the object, these three form features create the final geometry of the object. In the two FGM regions, FGM1 is a form feature while FGM2 is not. FGM2 does not satisfy the second condition of form features, i.e. FGM2, as a shape, does not contribute to the final part geometry of the exterior.

### 7.3.1.2 Material feature

Before we present the definition of material features, we first examine material variation in heterogeneous objects.



**Fig. 7.5 Features in a heterogeneous object**

Heterogeneous materials arise in materially optimized structures. They provide a smooth transition between different materials which are otherwise incompatible because of their different mechanical or chemical properties. The material variation usually corresponds to some particular functionality and design intent. They can be explicitly captured by a material volume, formally a material feature (28). Such a material volume can be represented in many different ways, e.g. a swept material volume (28) or a B-spline material volume (29).

A material feature is a region with some particular material composition function and this material function is not equal to the neighbouring volume's material functions. Such material composition variation is associated with some engineering significance, such as erosion protection, thermal balance, and biocompatibility.

A material feature is an enriched material volume. The relationship between a material feature and the material volume is the same as the relationship between a geometric feature and the geometric volume. The features contain engineering relevance while the volumes do not. Material features can be represented as a pair,  $MF(g, m)$ , where  $m$  has certain characteristics in the region  $g$  and is different from the material function elsewhere.

In this paper, when material functions are equal to each other for two regions  $(g_1, m_1)$  and  $(g_2, m_2)$ ,  $m_1 \equiv m_2$ , it means: (1) there is a  $C^\infty$  function  $m(x)$  for  $x \in g$ ,  $g = g_1 \cup g_2$ ; (2)  $m = m_1$  for  $x \in g_1$ ; and (3)  $m = m_2$  for  $x \in g_2$ .

The sample part in Fig. 7.5 has three material features: two FGM ( $Al_2O_3$ , Ceramic) region and one ceramic region.

### ***7.3.1.3 Observations on the form features and material features of heterogeneous objects***

The next issue to be examined is the relationship between form features and material features. Since we will define feature operations based on these features, it is important to determine what are the critical characteristics of these features. In the course of our investigation, we observe a number of significant points regarding the nature of these features.

*Observation 1:* Material features  $MF(g,m)$  form a partition of the part  $P$ . That is,

$$P = \bigcup_i MF_i$$

$$\overline{G} = g_1 | g_2 | \dots | g_n$$

Note,  $\overline{G}$  is defined as a closure in a three-dimensional manifold, and ‘|’ is a glueing operation.

Figure 7.6 shows the partition of the part geometry by material features. On the left is a complete geometry of the part shown in Fig. 7.5. On the right is a partition of the part volume. Each subvolume in the partition corresponds to one material feature in Fig. 7.5.

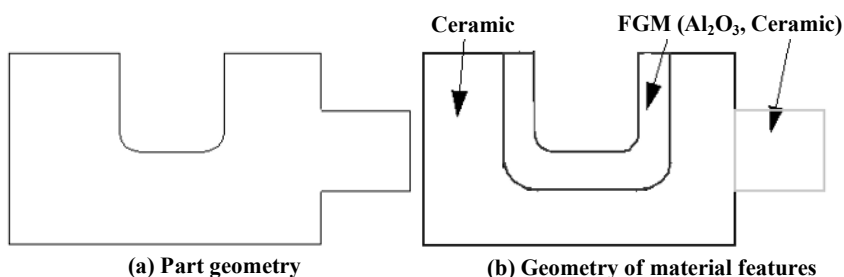
*Observation 2:* Form features form the geometry of part volume

$$\overline{G} = \prod_j FF_j$$

$\prod$  refers to the form feature operations, i.e. either an addition or a subtractive operation.

Figure 7.5b shows how form features form the part volume. Three features are added one by one and lead to the final part geometry.

*Observation 3:* The geometry volumes in form feature volumes and the material volumes in material features need not be identical.



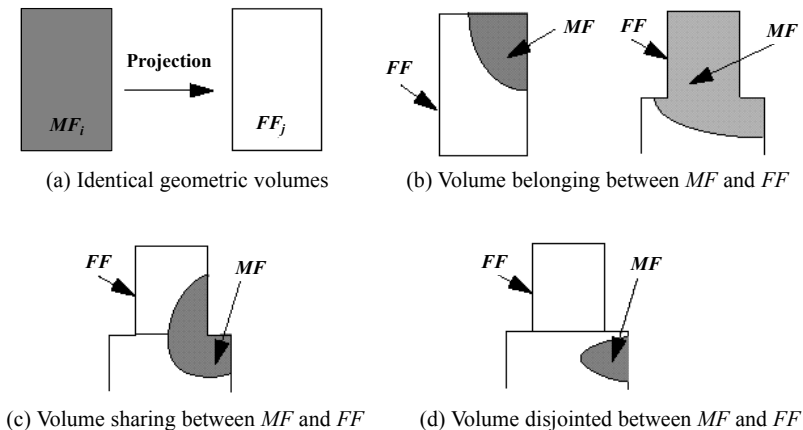
**Fig. 7.6** Material features partition part volume

In order to examine the relationships between form features and material features, we note the geometric volume of material feature  $MF_i$  as  $g(MF_i)$ , its operation with  $FF_j$  as  $g(MF_i) \otimes FF_j$ . It can be simplified to  $MF_i \otimes FF_j$ . The geometries of form features and material features have one of the following relationships (Fig. 7.7):

- $MF$  and  $FF$  have identical geometric volumes (*identical*)  
 $MF_i - FF_j = FF_j - MF_i = \emptyset$
- $MF$  belongs to  $FF$  or  $FF$  belongs to  $MF$  (*belonging*)  
 $MF \subset FF$  or  $FF \subset MF$
- $MF$  and  $FF$  share some subvolume (*sharing*)  
 $FF_i \cap MF_j \neq \emptyset$
- $MF$  and  $FF$  are disjointed (*disjointed*)  
 $FF_i \cap MF_j = \emptyset$

The above observations reveal that material features describe the part's interior material composition and form features describe the part's exterior geometric shape.

Even though using form features alone or material features alone may enable construction of the design model, using each type of feature alone is not sufficient to support the design process. Using form features alone, no partition of the part volume is obtained. Using material features alone, the design intent of the geometric features is



**Fig. 7.7 Relationship between form features and material features**

not captured. Often both form features and material features are necessary representations of the design intents. Therefore, feature-based design for heterogeneous objects needs to include both geometric and material features.

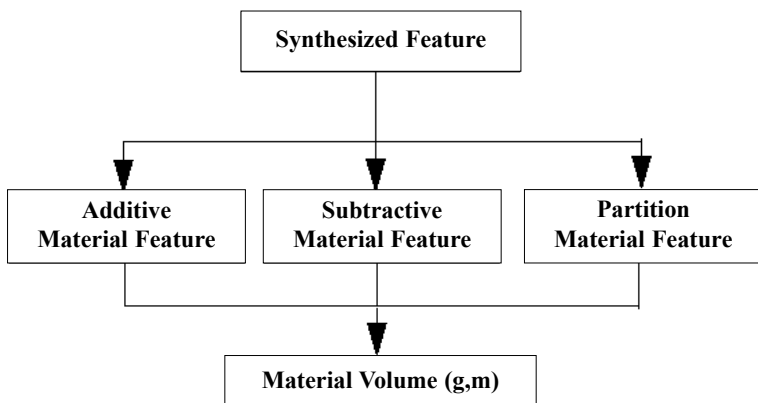
### 7.3.2 Synthesized features for feature-based constructive design

#### 7.3.2.1 *Synthesized features and semantics definition*

With our understanding of the relationships between the form features and material features, we can now proceed to the synthesis of form feature and material feature operations.

In STEP, the volume features are classified as additive and subtractive features. To be consistent with form feature classification in the STEP and the observed feature properties in heterogeneous objects, we propose the following feature operations in the context of heterogeneous object design: additive material feature, subtractive material feature, and partition material feature (Fig. 7.8). In responding to additive and subtractive features in STEP, we propose additive and subtractive material features. In responding to the partition properties of material features in heterogeneous objects, we propose partition material features. This classification is based on the modelling operation's impact on geometry.

These synthesized features support both form feature and material feature operations. It associates each material volume with one



**Fig. 7.8 A proposal for feature classification in heterogeneous objects**



geometric/material operator. They preclude redundant definition of the geometry in both form features and material features. The four types of relationships between form features and material features can be supported by the synthesized features using a constructive approach. In this approach, the building blocks are the synthesized features. The designer has two choices: either use the default materials to model form features and then partition the part volume with specific material composition functions, or glue a set of material feature volumes.

Before we present the details of the semantics definition for each feature operation, we will define some terms. For an object/region  $A(g, m)$ ,  $m(A)$  gives the material information  $m$ ,  $p(A)$  is the priority of the materials and it is useful when different materials are interacting with each other.

As noted before, ‘|’ is the aggregate/glueing operation. ‘|\*’ is the regularized glueing operation. For each face, if material functions over the face’s two adjacent regions are equal, the face shall be eliminated. That is,  $(g_1, m_1) |^* (g_2, m_2) = (g_1 \cup^* g_2, m_{12})$  when material function equality conditions are satisfied.

The three synthesized feature operations can be defined, respectively, as:

1. Additive material feature

$$(g_1, m_1) + (g_2, m_2) \\ = (g_1 - g_2, m_1) |^* (g_2 - g_1, m_2) |^* (g_1 \cap g_2, m_1 \otimes m_2)$$

2. Subtractive material feature

$$(g_1, m_1) - (g_2, m_2) = (g_1 - g_2, m_1)$$

3. Partition material feature

$$(g_1, m_1) / (g_2, m_2) = (g_1 - g_2, m_1) |^* (g_1 \cap g_2, m_1 \otimes m_2)$$

Figure 7.9 lists the three types of features and their semantics. Clearly, the part,  $C = A \otimes B$ , depends on the feature type (operation), and each region’s materials and the priority tag.

To resolve the material composition ambiguity over the intersection region, we introduce the material priority tag  $p$ , to each material volume. That is,

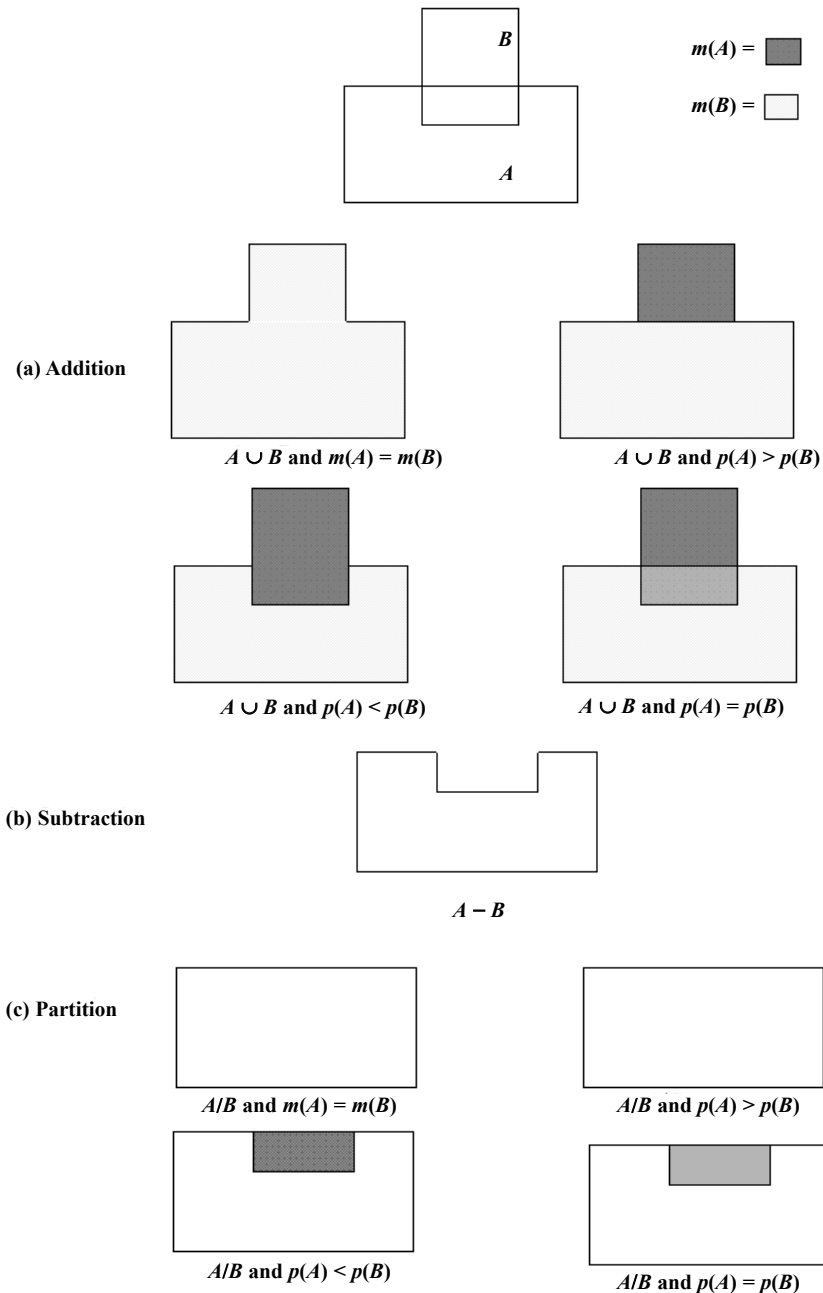


Fig. 7.9 Generic feature operations for heterogeneous objects

$$m_1 \otimes m_2 = \begin{cases} m_1, & \text{if } p_1 > p_2 \\ m_2, & \text{if } p_1 < p_2 \\ m_1 \oplus m_2, & \text{if } p_1 = p_2 \end{cases}$$

Note that here  $m_1 \oplus m_2$  is a user-defined function. It could be  $a_1 \cdot m_1 + (1 - a) \cdot m_2$ ,  $a \in (0,1)$ , or any other form. The  $m_1 \oplus m_2$  has been particularly useful for applications like doping, and implanting, where material volume is ‘contaminated’ by some exotic materials.

The material composition change during the synthesized feature operation is referred to as *material operation semantics*.

The partition features function in the same way as additive features over the intersection region ( $g_1 \cap g_2$ ), but they are not applicable to the region outside of  $g_1$ . This partition feature is used extensively for heterogeneous object modelling when material functions are imposed on a given geometry domain.

The three features provide a generic tool for heterogeneous object modelling. Many existing design/fabrication automation tools for heterogeneous objects processing are dedicated tools and they can be directly derived from the three synthesized features. For example, the feature operation semantics used in design by composition for layered manufacturing (30) and MEMS simulation (31) can all be derived from the synthesized features (17).

Based on these synthesized feature operations, a feature-based design methodology can be developed for heterogeneous object design. Such a methodology needs two enabling component techniques: how to combine material volumes and how to define material composition within each material volume. These two enabling techniques are presented in Sections 4 and 5, respectively.

## 7.4 CONSTRUCTIVE FEATURE OPERATIONS AND MATERIAL HETEROGENEITY MODELLING

### 7.4.1 Constructive feature operations through direct face neighbourhood alteration

Constructive feature operations need an effective modelling algorithm for combining the feature volumes. Given heterogeneous objects

$A = \{A_1 | A_2 | \dots | A_m\}$  and  $B = \{B_1 | B_2 | \dots | B_m\}$  and the feature operator, the resultant solid needs to be formed. It essentially includes two tasks:

- Determine the boundary of  $A$  and  $B$  that appears in the resultant solid  $C$  (*geometric boundary evaluation*)
- organize the resultant faces into regions and associate material function  $m_1$  to each region  $g_1$  (*material region forming*)

In this research, both the geometric boundary evaluation and material region forming are conducted based on a novel method, direct face neighbourhood alteration (32).

Neighbourhood is a well-known concept from topology (33). Direct face neighbourhood alteration is the core part of constructive operations for heterogeneous objects. In heterogeneous objects, each face has two neighbouring regions. We perceive the three-dimensional face's neighbourhood as a *two-sided face neighbourhood* and represent it as a combination of two one-sided face neighbourhoods from each adjacent region.

#### 7.4.1.1 One-sided face neighbourhood representation

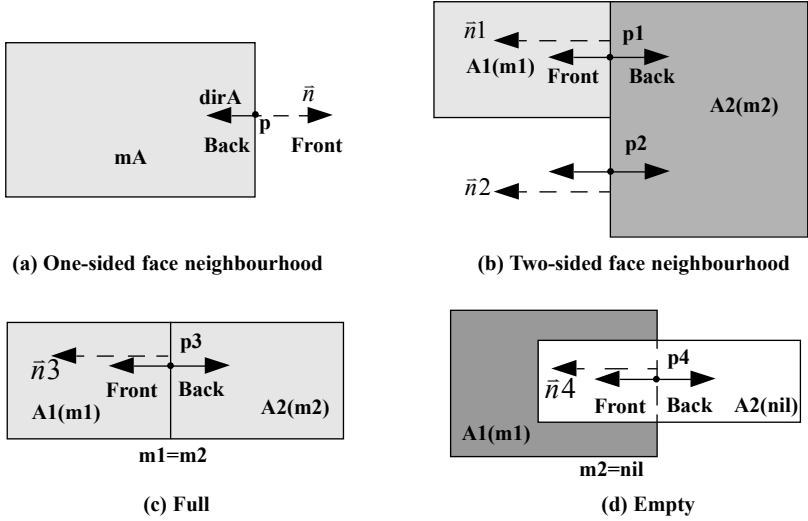
The face neighbourhood in each region is represented as a combination of normal direction of the face and material function of the region. Suppose point  $p$  lies on a face of region  $A$ , its neighbourhood is represented as:

$$nF_A = (dirA, mA) \quad (7.1)$$

Here the  $dirA$  is region  $A$ 's inward normal direction at point  $p$ ,  $mA$  is the material composition function in region  $A$ .

For example, in Fig. 7.10a, the point  $p$  in region  $A$ 's neighbourhood is  $nF(p) = (-n, mA)$ .

Denote the face's preserved reference normal direction at point  $p$  as  $n(\text{RefNormal})$ . The front side refers to the side of a face, which is in front of  $p$  along the normal direction  $n$ . The opposite side is called the back side. So each face has two one-sided neighbourhoods, respectively in two adjacent regions, i.e.  $nF_{\text{front}} = (\text{RefNormal}, m_{\text{front}})$ , and  $nF_{\text{back}} = (-\text{RefNormal}, m_{\text{back}})$ .



**Fig. 7.10 Three-dimensional face neighbourhood representation for a heterogeneous solid**

The three-dimensional face's complete neighbourhood representation at point  $p$  is a combination of  $nF_{\text{front}}$  and  $nF_{\text{back}}$ .

$$NF(p) = nF_{\text{front}} | nF_{\text{back}}$$

So the three-dimensional face's neighbourhood is a quadruple

$$NF(p, F) = (\text{RefNormal}, m_{\text{front}}) | (-\text{RefNormal}, m_{\text{back}}) \quad (7.2)$$

When both sides of a face have null material, the neighbourhood is *empty* and the face is in the exterior of the object. When both sides of a face have the same material function, the neighbourhood is *full* and the face is in the interior of a region. During the regularization process, faces with either empty or full neighbourhoods will be discarded.

For example, in Fig. 7.10b the two-sided face neighbourhood of points  $p1$  and  $p2$  are  $NF(p1) = (n1, m1) | (-n1, m2)$ ,  $NF(p2) = (n2, nil) | (-n2, m2)$ . In Figure 7.10c, the point  $p3$  has neighbourhood  $NF(p3) = (n3, m) | (-n3, m)$ . Therefore,  $p3$ 's neighbourhood is *full* and is completely interior to region  $B$ . In Fig. 7.10d, the point  $p4$  lies on the boundary of  $(A1-A2)$ . So its neighbourhood after the operation  $(A1-A2)$  is  $NF(p4) = (n4, nil) | (-n4, nil)$  and is *empty*.

### 7.4.1.2 Neighbourhood operations

Given the objects  $A$  and  $B$ , the faces from  $A$  and  $B$ , noted as  $F_A$  and  $F_B$ , can be classified against each other. There are five types of set membership classification (SMC) values:  $F_A$  in  $B$ ,  $F_A$  out  $B$ ,  $F_A$  on  $B/F_B$  on  $A$ ,  $F_B$  in  $A$ , and  $F_B$  out  $A$  (Fig. 7.11). Therefore, corresponding to the five SMC values, there are five  $NF$  operations for the operation  $A \otimes B$ : (1)  $NF_A \otimes B_j$  for  $F_A$  inside region  $B_j$ ; (2)  $A_i \otimes NF_B$  for  $F_B$  inside region  $A_i$ ; (3)  $NF_A \otimes NF_B$  for  $F_A$  and  $F_B$  that are co-faces; (4)  $NF_A \otimes B^C$  for  $F_A$  outside object  $B$ , i.e.  $F_A$  interacts with region  $B^C$ ; (5)  $A^C \otimes NF_B$  for  $F_B$  outside object  $A$ , i.e.  $F_B$  interacts with region  $A^C$ . Figure 7.11 shows the five neighbourhood operations. Since different regions have different material operation semantics, the  $NF$  operations are fulfilled by combining two separate  $nF$  operations, each of which operates according to the residing region's semantics.

The neighbourhood operation for  $F_A$  with region  $B_j$  can be represented as:

$$NF_A \otimes B_j = (nF_{A_{\text{front}}} \otimes B_j) | (nF_{A_{\text{back}}} \otimes B_j)$$

$nF_{A_{\text{front}}}$  and  $nF_{A_{\text{back}}}$  refer to face  $F_A$ 's front region and back region's neighbourhood. For the generality, one-sided face neighbourhood in

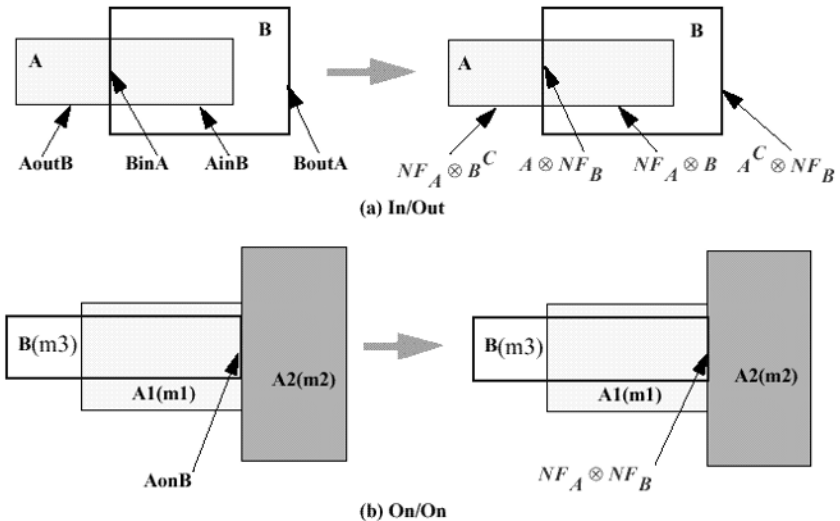


Fig. 7.11 Face membership classification and neighbourhood operation

region  $A_i$  is referred to as  $nF_{A_i}$ . The face neighbourhood for the object  $A$ 's complement set  $A^c$  is noted as  $nF_{A^c}$ .

An example of  $F_A$  interacting with region  $B$  is shown in Fig. 7.12 (bold line). From the four cases in the union operation, we have the following neighbourhood alteration rules:

$$nF_{A_i} \cup B_j = \begin{cases} nF_{A_i} & mA = mB \\ nF_{A_i} & pA > pB \\ (dirA_i, mB) & pA < pB \\ (dirA_i, mA \oplus mB) & pA = pB \end{cases} \quad (7.3)$$

The other types of face neighbourhood alteration can be derived similarly.

#### 7.4.1.3 Implementation of direct face neighbourhood alteration

A prototype system for feature-based constructive design based on the direct face neighbourhood alteration has been implemented using

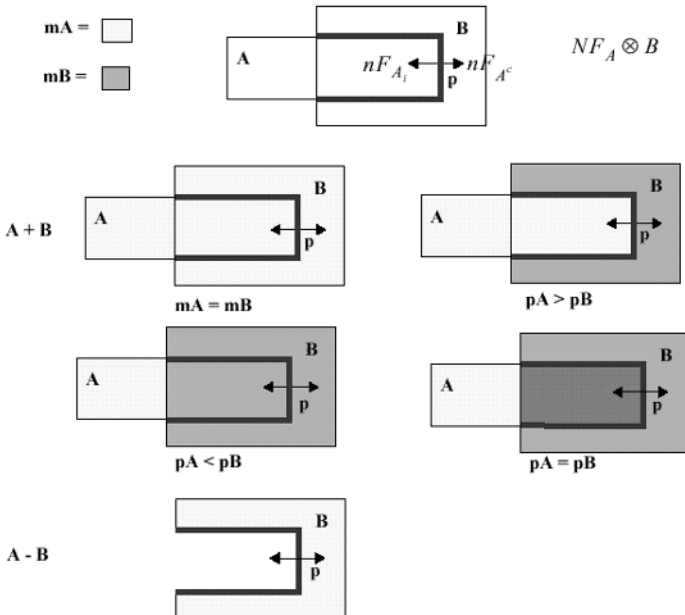
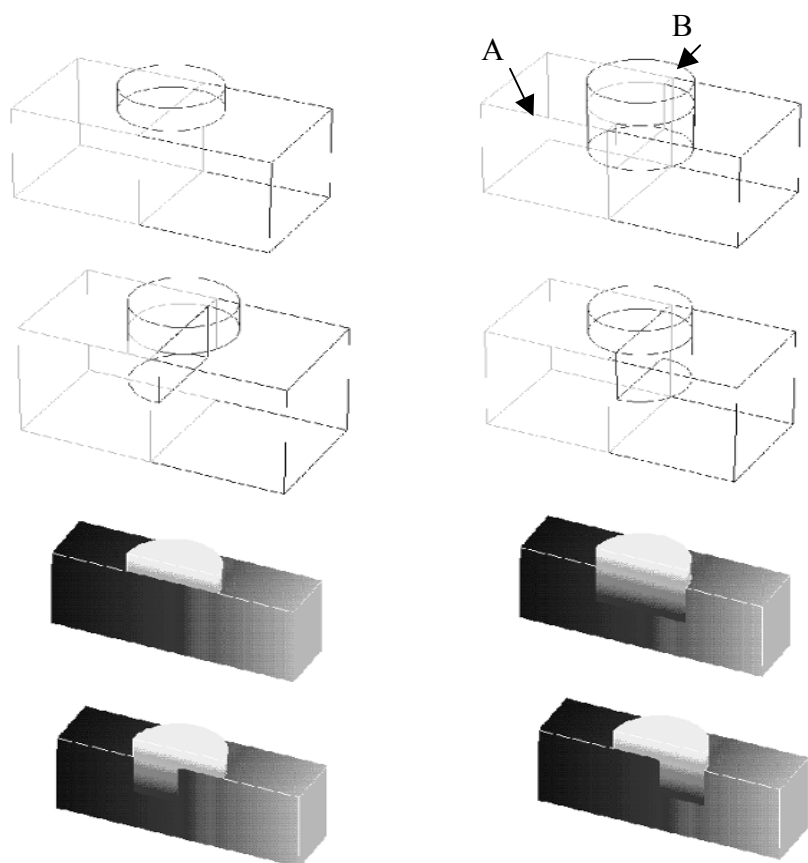


Fig. 7.12 Neighbourhood operations for  $F_A$  in  $B$

ACIS. Figure 7.13 shows a sample part, consisting of two feature volumes, *A* and *B*. By direct face neighbourhood alteration, the system gives different results, depending on the priority of each primitive. The bottom half of the figure is the shaded cross-section of the parts.

#### 7.4.2 Material heterogeneity specification for features/objects

In addition to the geometric modelling, material heterogeneity modelling is another important task in heterogeneous object design. In this section, we present the use of physics (diffusion)-based modelling to intuitively control material composition variation within each feature volume (17).



**Fig. 7.13** Sample part for face neighbourhood alteration



#### 7.4.2.1 B-spline tensor solid representation for heterogeneous objects

For each point  $(u,v,w)$  in the parametric domain of a tensor product B-spline volume  $V$ , there is a corresponding point  $V(u,v,w)$  at Cartesian co-ordinates  $(x,y,z)$  with material composition  $M$ , noted as  $(x,y,z,M)$ . We define such a B-spline volume as:

$$V(u,v,w) = \sum_{i=0}^n \sum_{j=0}^m \sum_{k=0}^l N_{i,p}(u) N_{j,q}(v) N_{k,r}(w) P_{i,j,k} \quad (7.4)$$

where  $P_{i,j,k} = (x_{i,j,k}, y_{i,j,k}, z_{i,j,k}, M_{i,j,k})$  are control points for the heterogeneous solid volume.  $N_{i,p}$ ,  $N_{j,q}$ , and  $N_{k,r}$  are the  $p$ th-degree,  $q$ th-degree, and  $r$ th-degree B-spline functions defined in the direction of  $u,v,w$ , respectively.

We can also have the B-spline representation for material properties:

$$E(u,v,w) = \sum_{i=0}^n \sum_{j=0}^m \sum_{k=0}^l N_{i,p}(u) N_{j,q}(v) N_{k,r}(w) E_{i,j,k} \quad (7.5)$$

where  $E_{i,j,k}$  is the material property at each control point. It can be obtained according to the volume fractions at each point.

#### 7.4.2.2 Diffusion-based modelling

In this section, we describe how the diffusion process generates different material composition profiles. Diffusion is a common physical process for the formation of material heterogeneity: in integrated circuit fabrication, in biological mass transport, in drug delivery from a polymer; the material composition variation can be described in most cases by diffusion.

The mathematical modelling of controlled material composition in these processes is based on Fick's laws of diffusion. Applying Fick's laws and using the divergence theorem, we have

$$\frac{dM}{dt} = Q + \frac{\partial}{\partial x_i} (D_{ij} \cdot \frac{\partial M}{\partial x_j}) \quad (7.6)$$

By the finite element approximation, it becomes

$$\left[ \int_{\Omega} \frac{\partial N_{\alpha}^m}{\partial x_i} \cdot D_{ij} \frac{\partial N_{\beta}^m}{\partial x_j} \right] M = \int_{\Omega} N_{\alpha}^m Q d\Omega - \int_{\Gamma_2} N_{\alpha}^m q_n d\Gamma \quad (7.7)$$

In matrix form, equation (7.7) becomes

$$KM = \bar{B} - \bar{S} \quad (7.8)$$

where

$$K_e = [k \int_{\Omega_e} \left( \frac{\partial N_i}{\partial x} \cdot \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \cdot \frac{\partial N_j}{\partial y} + \frac{\partial N_i}{\partial z} \cdot \frac{\partial N_j}{\partial z} \right) d\Omega] \quad (7.9)$$

$$\bar{B}_e = \left[ \int_{V_e} N^m Q dV \right], \quad \bar{S}_e = \left[ \int_{\Gamma_e} N^m q_n d\Gamma \right] \quad (7.10)$$

With function  $Q$  and  $q$  interpolated in terms of its nodal values, we have

$$\bar{B}_e = \left[ \int_{\Omega_e} N_i^m N_j^m d\Omega \right] Q_j$$

and

$$\bar{S}_e = \left[ \int_{\Gamma_e} N_i^m N_j^m d\Gamma \right] q_j$$

$K_e$  is the element stiffness matrix,  $\bar{B}_e$  is the element body force, and  $\bar{S}_e$  is the element surface force.

#### 7.4.2.3 Material heterogeneity specification

A prototype system for diffusion process-based material heterogeneity modelling is implemented on SUN Sparc workstations. The input of the system is a B-spline solid, consisting of a set of control points. The user interacts with the system in two ways. First, the user can change system parameters, such as  $Q$ , the material source (material/unit volume), and  $D$ , the material diffusion coefficient. Second, the user can impose constraints. The two types of interaction process continue until the user is satisfied with the result. When constraints are changed, the system matrices remain the same. Only when the system properties are changed should the system stiffness matrix and body force matrix be recalculated.

7.4.3 Example: feature-based design of a prosthesis

The following example of prosthesis design demonstrates such a feature-based design process.

Figure 7.14 shows a flowchart for the prosthesis design process. Starting from the design functions, users select materials and form the heterogeneous material features, each of which is a B-spline volume. The feature combination algorithm combines these features into a heterogeneous object. After the mechanical and biological properties are obtained from the database for each individual material, these properties at each point in this prosthesis can then be evaluated. If users are not satisfied with the properties, they can select new material for each volume or change volume fractions. These steps of changing material composition of each feature in the constructive process form a feature-based design process. After the property evaluation, property *in vitro* tests and animal tests are conducted before the designed prosthesis is used for medical purposes.

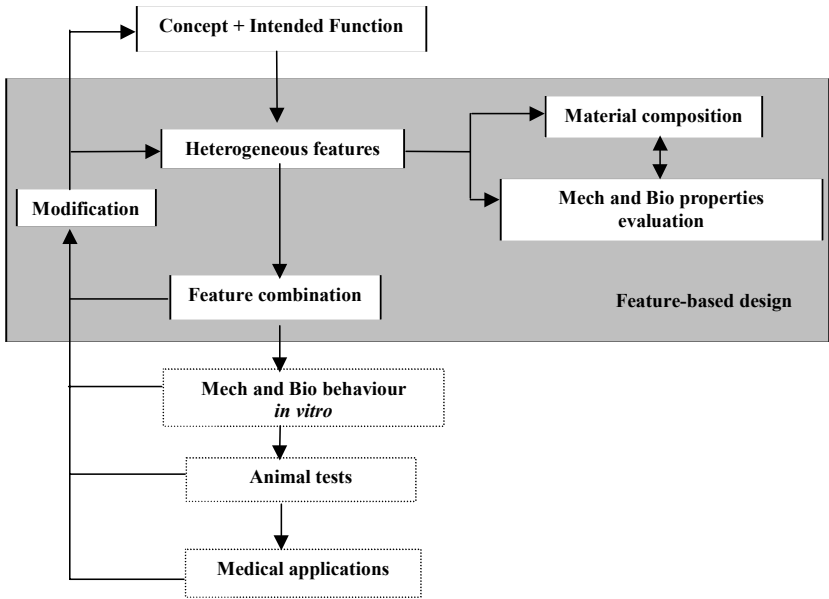
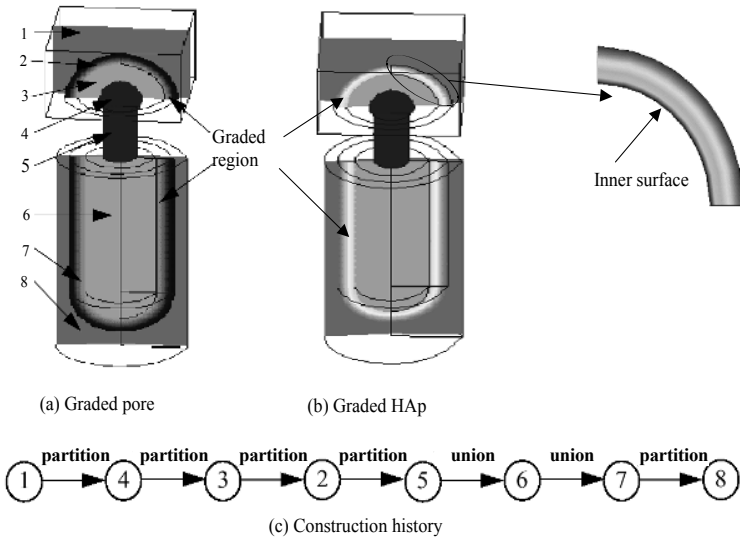


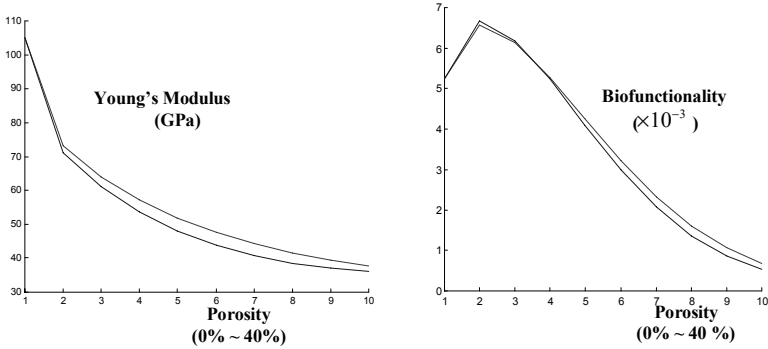
Fig. 7.14 Flowchart of a feature-based design process for a new prosthesis



**Fig. 7.15 Graded interface with a prosthesis**

Figure 7.15 shows a prosthesis designed following the flowchart in Fig. 7.14. The materials are titanium and graded HAp. Each of these design intents is represented as a separate B-spline volume (heterogeneous feature), such as in Regions 2 and 7 in Fig. 7.15. In these two regions, pore and HAp are modelled as one material, while the titanium is the other material. Regions 1 and 8 represent the bones. Regions 4 and 5 connect the two ends. Once the volume fraction for the pore and HAp is known, another fraction is used to separate the pore and HAp. This fraction is constant throughout the region. Figures 7.15a and b show the graded porous structure and graded HAp, respectively, with  $M_{\text{pore}}/M_{\text{HAp}} = 0.5$ . Figure 7.15c shows the construction history. The partition in the construction history is similar to union operation but with the intersection region's material redefined. Modification to the material composition can lead to different Young's modulus and biofunctionality (BF) distribution throughout the region. In Fig. 7.16, we show the properties variation due to the change of  $Q$  (material generation source). These values are measured at different distance points from the inner surfaces of the graded regions.

This example demonstrates that the feature-based design method not only provides an intuitive way to control the material compositions but



**Fig. 7.16 Variation of Young's modulus and biofunctionality due to  $Q$  change**

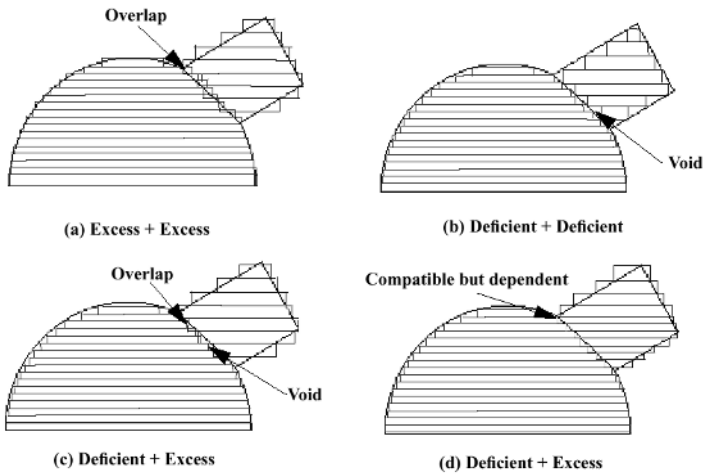
also provides means to directly control the material properties. This draws a distinction from existing design methods for the prosthesis design, where material composition design and material property evaluation are conducted separately and sequentially.

## 7.5 FEATURE-BASED FABRICATION IN LAYERED MANUFACTURING

In addition to facilitating the heterogeneous object design processes, the form features and material features in the heterogeneous object can also facilitate downstream applications. For example, material features can facilitate the material deposition process and form features can facilitate the numerically controlled (NC) machining and part assembly. This section details how features can facilitate the layer decomposition in layered manufacturing.

### 7.5.1 Staircase interaction in layered manufacturing

In order to increase the fabrication efficiency, feature-based fabrication localizes the geometric curvature effect and material gradation effect within each feature layer. However, the layer-wise deposition in layered manufacturing may lead to staircase interaction between neighbouring volumes. This staircase interaction results in geometric incompatibility. For example, in Fig. 7.17, a part, made of a hemisphere and a slant cylinder, is fabricated by feature-based slicing to decrease the build time (34). If both feature *A* and feature *B* are fabricated by excess deposition, the layers in *A* and *B* would interfere with each other (Fig. 7.17a). If both feature *A* and feature *B* are fab-



**Fig. 7.17 Staircase interaction**

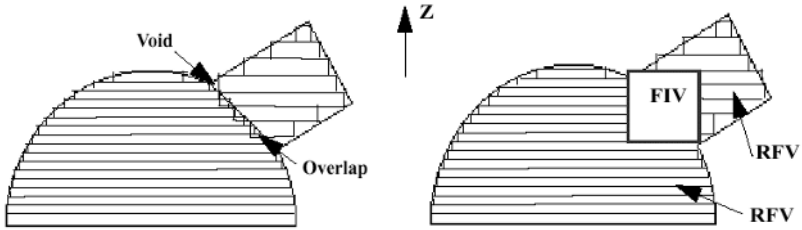
ricated by deficient deposition, a large void area is created between the neighbouring layers in *A* and *B* (Fig. 7.17b). If one is fabricated by excess deposition and the other by deficient deposition, the interaction typically results in both interference and void since the layer thickness of *A* and *B* do not match (Fig. 7.17c). The layers of *A* and *B* can become compatible only when the build direction of *A* is the same as *B* and the layer thickness of *A* and *B* are exactly the same (Fig. 7.17d). However, this compatibility is achieved by sacrificing the fabrication independence between neighbouring volumes. This kind of fabrication is equivalent to fabrication without volume decomposition.

### 7.5.2 Staircase interaction-free strategy

To eliminate the staircase interaction for each feature interaction, we propose the following two concepts:

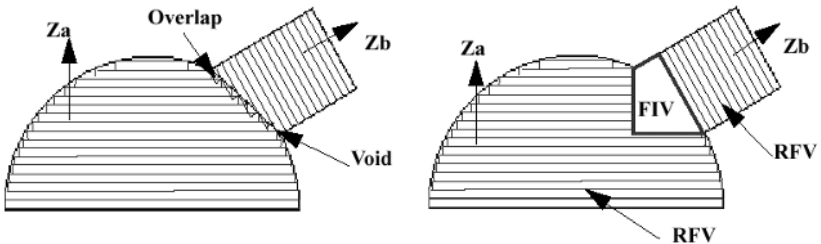
1. Feature interaction volume (FIV) is a transition volume formed to eliminate the staircase interaction between adjacent feature volumes.
2. Refined feature volume (RFV) is a feature volume devoid of all of its feature interaction volumes.

For example, in Fig. 7.18, features are to be fabricated along the vertical direction. An FIV is formed with the vertical and horizontal parting surfaces. This FIV eliminates the staircase interaction, and the adjacent RFVs can be fabricated independently and compatibly.



**Fig. 7.18 FIV eliminating staircase interaction**

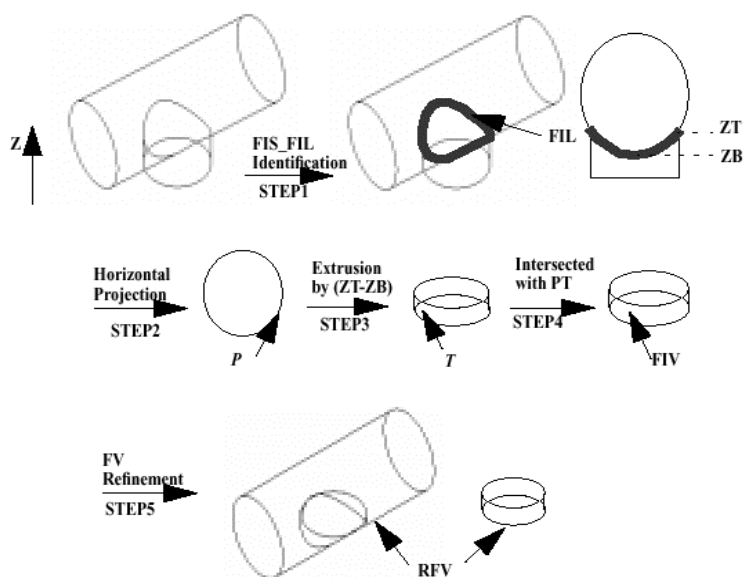
In Fig. 7.19, an FIV is generated for the features with different build directions. Still the separating surfaces of RFVs are either perpendicular or parallel to the build directions of the features. So there is no staircase interaction.



**Fig. 7.19 FIV eliminating staircase interaction for features with different build directions**

Figure 7.20 presents an example of feature volume decomposition for two intersecting cylinders. The processing of the feature interaction includes the following steps: (1) identifying the feature interaction surface and/or feature interaction loop; (2) obtaining the heights of the top point ( $ZT$ ) and bottom point ( $ZB$ ) at each feature interaction surface (FIS) and/or feature interaction loop (FIL); projecting the FIL/FIS on to a horizontal plane to get the projection  $P$ ; (3) extruding the projection  $P$  from  $ZB$  to  $ZT$ ; (4) intersecting the extruded volume  $T$  with the part volume to get the feature interaction volume; (5) generating the refined feature volume.

The details of the algorithm and the properties of FIVs are discussed in reference (34).



**Fig. 7.20 Processing of feature interaction**

### 7.5.3 Implementation

#### 7.5.3.1 Compatible volume decomposition

With the above feature interaction processing, we developed a system that can decompose the parts into a set of volumes, from which compatible deposition can be achieved. That is, there is no staircase interaction between the neighbouring volumes and the slicing of the feature volumes are independent.

Figure 7.21 shows a volume decomposition of the prosthesis in different build directions. With the orientation as in Fig. 7.21a, there is no material gradation effect. With the orientation as in Fig. 7.21b, the volumes are decomposed in a way that the curvature effect and material gradation effect are localized with each feature.

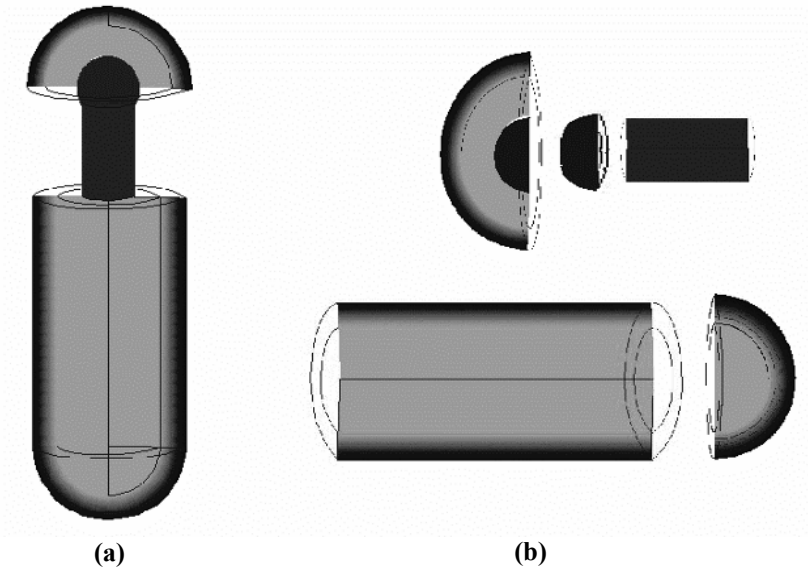
In order to experimentally validate the effectiveness of feature-based fabrication in resolving the staircase interaction, we devised our experiments on the Stratsys FDM machine. The dimensional and strength test experiments indicate that compatible deposition by the vertical parting surface introduced by FIV gives better dimensional accuracy and surface quality, and higher strength.



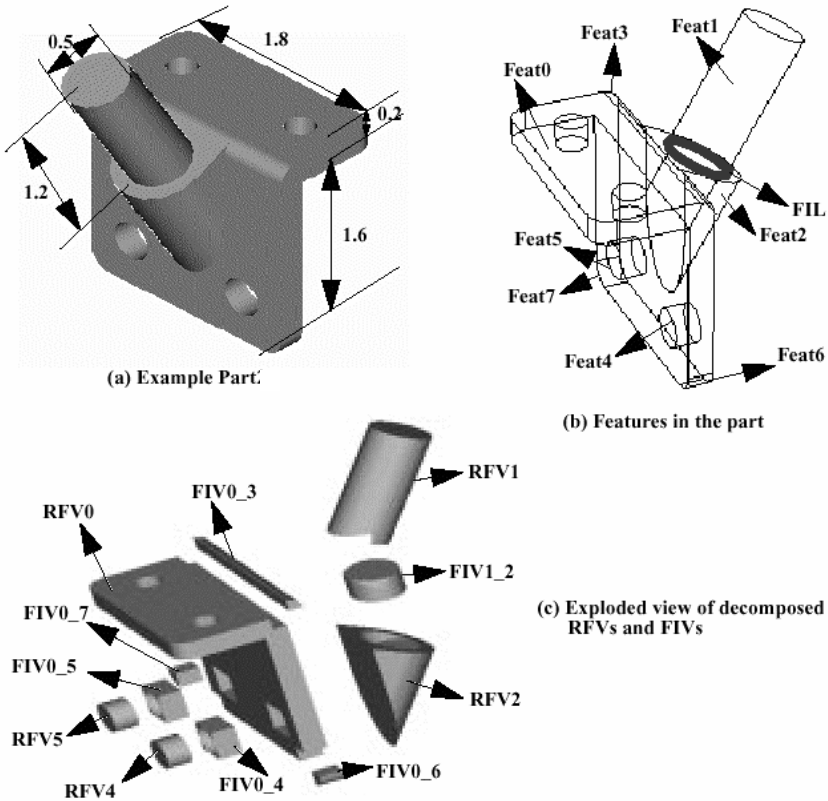
### 7.5.3.2 Build-time saving using feature-based fabrication

A build-time comparison study is conducted to validate the build-time efficiency using feature-based fabrication method. In the example part of Fig. 7.22, the sample part (downloaded from NIST repository (35) with a scale 0.02) is also sliced by the adaptive slicing, local adaptive slicing, and feature-based slicing methods. For this part, there are FIVs generated by the volume decomposition algorithm. Figure 7.22b shows a list of those features in the example part that have FIVs. An FIV is generated for each feature interaction and the feature volumes are refined. For instance, the bold dark line in Fig. 7.22b is an FIL between Feat1 and Feat2. FIV1\_2 is then created for the feature interaction between Feat1 and Feat2 as shown in Fig. 7.22c. After the FIV is generated, the features are refined and RFV1 and RFV2 are obtained respectively from Feat1 and Feat2.

The features in Fig. 7.22 fall into three categories: additive features (Feat0, Feat1, and Feat2), subtractive features (Feat4 and Feat5), and surface features (Feat3, Feat6, and Feat7). Correspondingly, the protrusion features have RFVs (RFV0, RFV1, and RFV2) made of part material; depression features have RFVs (RFV4 and RFV5) made of



**Fig. 7.21 Compatible volume decomposition**



**Fig. 7.22 Example (with FIVs)**

sacrificial support material; and surface features have no RFVs. All the FIVs are made of part material (FIV1\_2, FIV0\_3, FIV0\_4, FIV0\_5, FIV0\_6, and FIV0\_7).

After all the volumes are obtained, adaptive slicing is done separately for each volume. In this example, the cusp height for the part is 0.005 in. The minimum layer thickness is 0.002 in and the maximum layer thickness is 0.015 in. The boundary box of this part is 2.8 in  $\times$  1.8 in  $\times$  2.7 in. The key dimensions of the part are shown in Fig. 7.22a. The time comparison is shown in table 7.1. Note, the build time for the underneath support structure is approximately 1.6 h and is not included in the time shown in the table. Feature-based slicing yields a saving of 27 per cent build time when compared to adaptive slicing methods and local adaptive slicing methods (36).

Table 7.1 Time comparison for various methods

Methods	Time (h)
Adaptive slicing	3.8
Local adaptive slicing	3.8
Feature-based slicing	3

7.6 CONCLUSION AND FUTURE WORK

7.6.1 Conclusion

In the context of heterogeneous objects, we propose the use of features to facilitate the high level (explicit) conceptualizing of material composition and gradation and its downstream transforming to the fabrication. Based on our examination of the relationships between the form features and the material features, a feature-based design methodology is developed for heterogeneous object design. It is a constructive design process based on a set of user-predefined heterogeneous features. To improve the efficiency of such constructive feature operations, a direct face neighbourhood alteration method is developed. To model material heterogeneity effectively and efficiently within each feature, a physics-based B-spline heterogeneous object modelling method is researched and developed. In this method, B-spline representation is utilized to increase model coverage, and a physics process (diffusion process) is used to generate material composition profile to increase operation convenience.

This chapter also presents an effective feature-based fabrication methodology for LM. Such a method is developed to resolve the inherent dilemma in layered manufacturing – conflicting requirements of low build time and high surface quality. Feature-based fabrication effectively handles this issue. It localizes the curvature effect and material gradation effect within each feature layer. A new concept – *feature interaction volume* – is introduced to eliminate the staircase interaction between the neighbouring fabrication volumes. Experimental results and quantitative analysis demonstrate that feature-based fabrication saves build time and gives better overall surface quality, dimensional accuracy, and material strength.

### 7.6.2 Future work

This research has addressed two important issues in heterogeneous object realization – design methodology for heterogeneous objects and fabrication methodology in layered manufacturing. The contributions resulting from this research will impact upon the future research in the field of heterogeneous object realization. This section lists some future research areas based on these contributions. Some of the following topics are direct extensions of the ideas presented in this work, while others are generalizations of the concepts that are applied in this work. These topics include: design feature interaction, development of a feature base for heterogeneous object design, feature recognition from heterogeneous object model, applications of time-dependent heterogeneous objects in the biomedical field, design/analysis integration, and process planning for the fabrication of heterogeneous objects.

**Design feature interaction:** In this work, when features interact with each other during the design process, three operations (additive, subtractive, and partition) are used. At the feature interaction regions, users have to explicitly specify material composition. It may be better to have automatic material survival rules at those design feature interaction regions. This study of material feature interaction can be greatly enhanced by a more detailed understanding of how physically different materials can be synthesized together and the resulting material properties.

**Feature base for heterogeneous object design:** A generic feature-based design methodology is presented in this chapter. Applications based on this methodology can be developed for many fields, such as prosthesis design and cutting tools design. Each specific application requires a dedicated design feature base. Building such feature bases needs design knowledge and experience in each specific field.

**Feature recognition from heterogeneous object model:** This chapter presents a feature-based fabrication methodology for layered manufacturing. Explicit feature information (both geometry and material composition) can facilitate process planning for layered manufacturing. In this work, feature information is assumed to be given. For those heterogeneous object models where explicit features are not given, a feature recognition module is necessary.

**Applications of time-dependent heterogeneous objects:** Based on the physics-based B-spline heterogeneous object modelling method, a time-dependent heterogeneous object modelling system can be derived. Existing modelling methods only model the geometric deformation. The work in this research opens a new chapter in volume modelling. It models the dynamic material variation. This method can be used in applications where the material composition changes over time, particularly in biomedical/diagnosis fields, such as biodegradation.

**Design/analysis integration:** The physics-based B-spline heterogeneous object modelling method uses the finite element method to calculate the material composition at each control point of the B-spline volume. That is to say, each B-spline volume has already had discretized elements. Therefore, transferring such a model for analysis precludes the need for meshing. In the meantime, such B-spline representation has also been extended to represent material properties. Therefore, an integrated design/analysis system can be derived based on this work without a meshing process.

**Process planning for the fabrication of heterogeneous objects:** This chapter only presents a methodology in layer decomposition to resolve the conflicting requirements between the surface quality and build time. The other process planning tasks for layered manufacturing, such as orientation selection, support creation, and tool path generation, have to be addressed in the context of heterogeneous object fabrication. Existing methods have only addressed the issues in the fabrication of homogeneous objects. The variation in materials in heterogeneous objects creates a new dimension to these problems.

## ACKNOWLEDGEMENTS

We gratefully acknowledge the financial support from NSF (grant MIP-9714751).

## REFERENCES

- 1 Grayer, A. (1976) A computer link between design and manufacture. PhD dissertation, University of Cambridge, September.
- 2 Shah, J. J., Mantyla, M., and Nau, D. S. (Editors) (1994)

*Advances in Feature Based Manufacturing* (Elsevier).

- 3 Qian, X. and Dutta, D. (1988) Interoperability of layered manufacturing data: LMDData. ASME DETC, Atlanta, Georgia.
- 4 Requicha, A. A. (1980) Representations for rigid solids: theory, methods and systems. *Computing Survey*, **12**(4).
- 5 Hoffmann, C. M. (1989) *Geometric and Solid Modeling: An Introduction* (Morgan Kaufmann Publishers).
- 6 Weiler, K. J. (1986) Topological structures for geometric modeling. PhD dissertation, Rennselaer Polytechnic Institute, New York.
- 7 Pratt, M. (1988) Synthesis of an optimal approach to form feature modeling. In *Proceedings of the ASME Conference on Computers in Engineering*, San Francisco, August.
- 8 Rossignac, J. and O'Connor, M. (1989) SGC: a dimension-independent model for pointsets with internal structures and incomplete boundaries. In *Geometric Modeling for Product Engineering* (Eds M. Wozny, J. Turner, and K. Periss), pp.145–180 (IFIP, Amsterdam: Elsevier, North-Holland).
- 9 Rossignac, J. and Requicha, A. A. G. (1991) Constructive non-regularized geometry. *Computer-Aided Design*, **23**(1), 21–32.
- 10 Middleditch, A. E., Reade, C. M. P., and Gomes, A. J. (1999) Set-combinations of the mixed dimensional cellular objects of the Djinn API. *Computer-Aided Design*, **31**, 683–694.
- 11 Kumar, V. and Dutta, D. (1998) An approach to modeling and representation of heterogeneous objects. *ASME Journal of Mechanical Design*, **120**(4), December.
- 12 Jackson, T. (2000) Analysis of functionally graded material object representation methods. PhD dissertation, MIT, February.
- 13 Jackson, T., Liu, H., Patrikalakis, N. M., Sachs, E. M., and Cima, M. J. (1999) Modeling and designing functionally graded material components for fabrication with local composition control. *Materials and Design*, special issue (Elsevier Science, The Netherlands).
- 14 Wu, Z., Soon, S. H., and Lin, F. (1999) NURBS-based volume modeling. *International Workshop on Volume Graphics*, pp. 321–330.
- 15 Rvachev, V. L., Sheiko, T. I., Shaipro, V., and Tsukanov, I. (2000) Transfinite interpolation over implicitly defined sets. Technical report SAL2000-1, Spatial Automation Laboratory, University of Wisconsin-Madison.

- 16 Park, S.-M., Crawford, R. H., and Beaman, J. J. (2000) Functionally gradient material representation by volumetric multi-texturing for solid freeform fabrication. In 11th Annual Solid Freeform Fabrication Symposium, Austin, Texas, 7–9 August.
- 17 Qian, X. (2001) Feature methodologies for heterogeneous object realization. PhD dissertation, Mechanical Engineering, University of Michigan, April.
- 18 Marsan, A. and Dutta, D. (1998) On the application of tensor product solids in heterogeneous solid modeling. In ASME Design Automation Conference, Atlanta, Georgia, September.
- 19 Radeva, P., Amini, A. A. and Huang, J. (1997) Deformable B-solids and implicit snakes for 3D localization and tracking of SPAMM MRI data. *Computer Vision and Image Understanding*, **66**(2), 163–178.
- 20 Bendose, M. P. and Kikuchi, N. (1988) Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, **71**, 197–224.
- 21 Cherkaev, A. and Kohn, R. (1997) *Topics in the Mathematical Modelling of Composite Materials* (Birkhauser).
- 22 Dutta, D., Prinz, F. B., Rosen, D., and Weiss, L. (2001) Layered manufacturing: current status and future trends. *ASME Journal of Computing and Information Science in Engineering*, **1**, 60–71.
- 23 Kulkarni, P. and Dutta, D. (1996) An accurate slicing procedure for layered manufacturing. *Computer-Aided Design*, **28**(9), 683–697.
- 24 Suh, Y. S. and Wozny, M. J. (1994) Adaptive slicing of solid freeform fabrication processes. In Proceedings of Solid Freeform Fabrication Symposium, The University of Texas at Austin, 8–10 August, pp. 404–411.
- 25 Dolenc, A. and Makela, I. (1994) Slicing procedure for layered manufacturing techniques. *Computer-Aided Design*, **26**(2), February, 119–126.
- 26 Sabourin, E., Houser, S. A., and Bohn, J. H. (1997) Accurate exterior, fast interior layered manufacturing. *Rapid Prototyping Journal*, **3**(2), 44–52. Thesis: <http://www.cadlab.vt.edu/bohn/projects/sabourin>
- 27 Kumar, V. (1999) Solid modeling and algorithms for

- heterogeneous objects. PhD thesis, Mechanical Engineering, University of Michigan.
- 28 Qian, X. and Dutta, D. (1998) Features in the layered manufacturing of heterogeneous objects. Symposium of Solid Freeform Fabrication, The University of Texas at Austin, pp. 689–696, August.
  - 29 Qian, X. and Dutta, D. (2001) Physics based modeling for heterogeneous objects. *ASME Transactions Journal of Biomechanical Engineering*, submitted April.
  - 30 Binnard, M. and Cutkosky, M. R. (2000) A design by composition approach for layered manufacturing. *ASME Transactions, Journal of Mechanical Design*, **122**(1), March, 91–101.
  - 31 Dixit, H., Taylor, D., and Kannapan, S. (1997) 3D geometric simulation of MEMS fabrication processes: a semantic approach. In Proceedings of Fourth Symposium on *Solid Modeling and Applications*, Atlanta, Georgia.
  - 32 Qian, X. and Dutta, D. (2000) Heterogeneous object modeling through direct face neighborhood alteration. *ACM Transactions on Graphics* (in preparation).
  - 33 Armstrong, M. A. (1983) *Basic Topology* (Springer-Verlag, New York).
  - 34 Qian, X. and Dutta, D. (2001) Feature based fabrication in layered manufacturing. *ASME Transactions Journal of Mechanical Design* **123**(3), September, 337–345.
  - 35 NIST Design, Processing Planning and Assembly Repository, <http://www.mel.nist.gov/pptb>
  - 36 Tyberg, J. and Bohn, J. H. (1998) Local adaptive slicing. *Rapid Prototyping Journal*, **4**(3), 118–127. Thesis: <http://www.cadlab.vt.edu/bohn/projects/tyberg>



*This page intentionally left blank*

# Chapter 8

## CAD modelling and slicing of heterogeneous objects for layered manufacturing

**Y K Siu and S T Tan**

Department of Mechanical Engineering, The University of Hong Kong

### 8.1 INTRODUCTION

Recent studies in materials processing and engineering have led to a new class of material called ‘functionally graded materials’ (FGM). FGM forms an important new area of materials research in which two materials with very different properties are connected by a region where the material properties may vary smoothly with position. These innovative FGMs can be engineered to achieve improvements in the mechanical properties to the designed component. FGMs are distinguished from isotropic materials by the gradients of composition, phase distribution, texture, and related functional properties.

S. Suresh and A. Mortensen in their book (1) discuss the benefits of applying FGM in engineering designs. They reach several conclusions:

- By avoiding a sharp change in material compositions, the problem of thermal mismatch across the material domains can be minimized so that the induced thermal stresses will be suppressed.
- Gradual variation in material compositions can prevent severe stress concentrations and singularities at material interfacial regions.

- For some dissimilar solids such as a metal and a ceramic, the interfacial strength between the two solids can be increased by introducing a continuous variation in composition across the material domains.

The term ‘heterogeneous object’ refers to the inhomogeneity of material composition in the object. Heterogeneous objects are generally classified as objects with clear material domains and those without clear material domains (i.e. FGM). Current development of FGMs in CAD modelling systems focuses on representing and controlling local material composition (2–5) within the object boundary. Kumar and Dutta (2) proposed a representation scheme for decomposing the object into multiple cells called  $r_m$ -objects and each cell contains a kind of material or material grading. Jackson *et al.* (4) had also proposed a representation scheme for capturing the material composition blending by subdividing a model into subregions. For each subregion, a blending function is used to define the material composition. Chiu and Tan (5) elaborated on a scheme for building a material tree of an object into the Unigraphics CAD system. By extracting information from this material tree and generating a modified STL file format, a multiple materials object was built by a fused deposition modelling (FDM) machine.

## 8.2 HETEROGENEOUS SOLID MODELLING

To model a heterogeneous object with material variations, a CAD system should first define the material composition of each point inside the model. From the Euclidean space  $E^3$ , a bounded region  $A^3$  is defined for denoting the geometry of an object:

$$A^3 = \{\underline{x} \in E^3 \mid \underline{x} \text{ inside the object}\} \quad (8.1)$$

where  $\underline{x}$  = a point in  $E^3$

### 8.2.1 A material composition array $M$

A material composition array  $M$  is defined for mapping the material composition information to every point of the object. It is defined that each element of the material composition array  $M$  represents the volume fraction of a predefined primary material and the total volume fraction of the primary materials for the material composition arrays should be summed to one.

$$\sum_{j=1}^n M_j = 1 \quad (8.2)$$

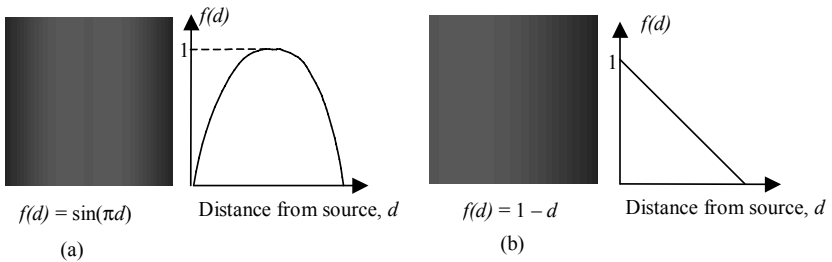
where

$M_j$  =  $j$ th element of the material composition array  $M$   
 $n$  = number of primary materials

For example,  $M = (0\%, 30\%, 20\%, 50\%)$  stands for a given point  $\underline{x}$ , the material composition of  $\underline{x}$  is 0 per cent material A, 30 per cent material B, 20 per cent material C, and 50 per cent material D.

### 8.2.2 Material grading function $f(d)$

The material grading function  $f(d)$  is a function of distance from the geometric point to a reference datum. The reference could be a fixed reference in the co-ordinate system of the model, such as a point, a line, a plane, or the outer most boundaries of a model. The  $f(d)$  can be either a linear or a non-linear mathematical function. Figures 8.1a and b show the effects of grading using different grading functions.



**Fig. 8.1 (a) material grading using a non-linear function.  
 (b) material grading using a linear function**

In addition, the volume fraction for the  $j$ th primary material in  $A^3$  is defined by:

$$\begin{aligned} v_j &= M_{sj} & \text{for } f(d) < 0 \\ v_j &= f(d) \times (M_{ej} - M_{sj}) + M_{sj} & \text{for } 0 \leq f(d) \leq 1 \\ v_j &= M_{ej} & \text{for } f(d) > 1 \end{aligned} \quad (8.3)$$

where

$M_{sj}$  = volume fraction of  $j$ th element of the material composition array  $M_s$

$M_{ej}$  = volume fraction of  $j$ th element of the material composition array  $M_e$

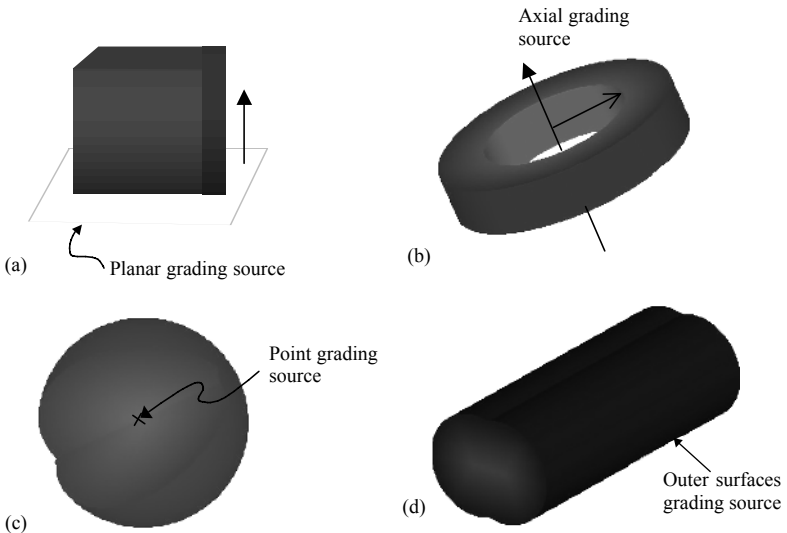
$v_j$  = the volume fraction for the  $j$ th primary material during grading

$$\sum_{j=1}^n v_j \equiv \sum_{j=1}^n [f(d) \times (M_{ej} - M_{sj}) + M_{sj}] = 1 \quad (8.4)$$

Figure 8.2 shows the effects of grading and the definition of  $d$  in the grading function  $f(d)$  in the different types of grading sources.

### 8.2.3 Grading source

The grading source is the ‘origin of grading’ and gathers all the necessary information such as reference datum, material composition arrays, and material grading functions. The reference datum in the grading source could be a fixed reference in the co-ordinate system of



**Fig. 8.2 (a)  $d$  = distance of a point from a reference plane.**

**(b)  $d$  = radial distance of a point from a reference axis.**

**(c)  $d$  = distance of a point from a reference point.**

**(d)  $d$  = minimum distance of a point from the selected outer surfaces**

the model, such as a point, line, or plane. Once a grading source is assigned to a reference, the Euclidean space  $E^3$  is fully occupied by the grading source and three grading regions are then generated. Material grading, however, will only occur in the intersection between the object and the grading sources. Three grading regions are defined according to the material distribution function  $f(d)$ . A point in  $A^3$  can fall in one and only one of the following regions corresponding to the grading source  $G$ .

1. Effective grading region,  $G_{\text{eff}}^3 = \{\underline{x} \in A^3 \mid 0 \leq f(d) \leq 1\}$

It has a valid material composition function range ( $0 \leq f(d) \leq 1$ ), i.e. it is inside the effective grading region.

2. -ve complementary region,  $G_{-c}^3 = \{\underline{x} \in A^3 \mid f(d) < 0\}$

The material distribution function  $f(d)$  of the geometric point is smaller than zero, i.e. it is inside the -ve complementary region.

3. +ve complementary region,  $G_{+c}^3 = \{\underline{x} \in A^3 \mid f(d) > 1\}$

The material distribution function  $f(d)$  of the geometric point is greater than one, i.e. it is assigned as the +ve complementary region.

Therefore, for each grading source  $G$ , we can pair up the geometric grading regions and the material composition arrays of these regions as shown in Fig. 8.30.

$$G = ((G_{\text{eff}}^3, B_{\text{eff}}), (G_{-c}^3, B_{-c}), (G_{+c}^3, B_{+c})) \quad (8.5)$$

where

$B_{\text{eff}}$  denotes the material distribution in the effective grading region

$B_{-c}$  denotes the material distribution in the -ve complementary region

$B_{+c}$  denotes the material distribution in the +ve complementary region,

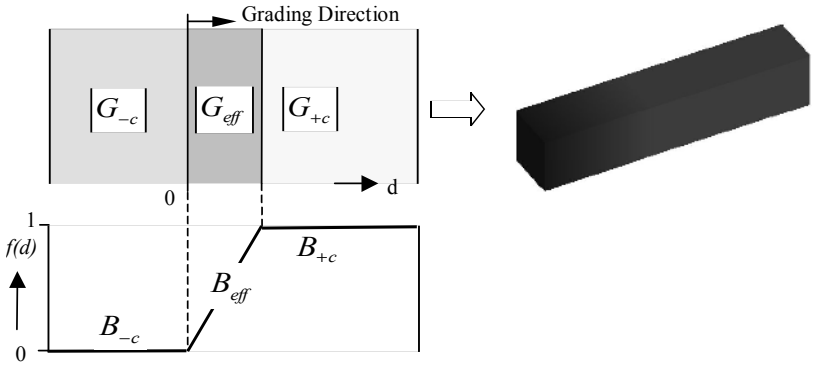
The material composition of each point in the heterogeneous object can be described by a material composition array and the following equations define the material composition distribution of the three regions respectively.

$$B_{\text{eff}} = f(d) \times (M_e - M_s) + M_s \quad (8.6)$$

$$B_{-c} = M_s \rightarrow \quad (8.7)$$

$$B_{+c} = M_e \quad (8.8)$$

where  $M_s$  and  $M_e$  are the two end material composition arrays in the material grading.



**Fig. 8.3 Model of a heterogeneous object with a planar grading source**

For a given heterogeneous object  $H$ , a source list  $G_s$  is defined and those grading sources that are associated with this object are members of this list.

$$G_s = \{G_i \mid G_i \text{ is associated with the object } H, i = 1 \dots n\} \quad (8.9)$$

where

$n$  is the total number of grading sources

$G_i$  is the  $i$ th grading source in the heterogeneous object

Therefore, a heterogeneous object  $H$  can be defined as  $H = (A^3, G_s)$ .

More than one grading source can be associated with a heterogeneous object, for example,  $G_s = (G_1, G_2)$  may introduce some intersecting regions between the grading sources. To determine the material compositions of the intersecting regions, a dominant factor  $r$  and a weighting ratio  $w$  are included in each grading source.

For example, consider two grading sources  $G_1$  and  $G_2$ ,

$$G_1 = ((G_{\text{eff}1}^3, B_{\text{eff}1}), (G_{-c1}^3, B_{-c1}), (G_{+c1}^3, B_{+c1}))$$

$$G_2 = (G_{\text{eff}2}^3, B_{\text{eff}2}), (G_{-c2}^3, B_{-c2}), (G_{+c2}^3, B_{+c2}))$$

The dominant factors  $r_1, r_2$  and weighting ratios  $w_1, w_2$  are defined with respect to the grading sources. Table 8.1 shows the material composition of the effective grading regions upon intersection. It should be noted that the effective grading region is always dominant over the complementary grading regions when overlapping of these regions occurs.

**Table 1: Material composition at intersecting regions**

Region	Material composition	
$G_{\text{eff}}^3 \cap^* G_{-c}^3$	$B_{\text{eff}}$	
$G_{\text{eff}}^3 \cap^* G_{+c}^3$	$B_{\text{eff}}$	
$G_{\text{eff}1}^3 \cap^* G_{\text{eff}2}^3$	If $r_1 > r_2$	$B_{\text{eff}1}$
	If $r_1 < r_2$	$B_{\text{eff}2}$
	If $r_1 = r_2$	$B_{\text{eff}1} \otimes_{w_1 w_2} B_{\text{eff}2}$

$\cap^*$  denotes the regularized intersection operation.

An intersecting operator ' $\otimes$ ' is defined to sum up the material compositions of these intersecting grading sources. The volume fraction of  $j$ th material ( $M_j$ ) in the intersecting regions is determined according to their respective weighting ratios:

$$B_{\text{eff}1} \otimes_{w_1 w_2} B_{\text{eff}2} \dots \otimes_{w_{(m-1)} w_m} B_{\text{eff}m} = \frac{w_1 B_{\text{eff}1} + w_2 B_{\text{eff}2} + \dots + w_m B_{\text{eff}m}}{\sum_{i=1}^m w_i} \quad (8.10)$$

where

$m$  = number of effective grading regions associated to the grading region

$B_{\text{eff}(i)}$  = material distribution of  $i$ th grading source

$w_i$  = weighting ratio of  $i$ th grading source



and

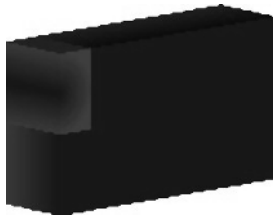
$$\sum_{i=1}^m \sum_{j=1}^n \frac{w_i B_{ij}}{\sum_{i=1}^m w_i} = 1 \quad (8.11)$$

where

$B_{ij}$  = volume fraction of  $j$ th primary material in the  $i$ th grading source

$n$  = number of materials

Besides associating different grading sources within a heterogeneous solid model, different heterogeneous solid models having their own grading sources list can also be modelled together as if they are of separate entities as shown in Fig. 8.4.



**Fig. 8.4 Two heterogeneous solid models with axial and planar grading source respectively without interfering with each other**

## 8.3 DISCUSSION ON EXAMPLE CASES

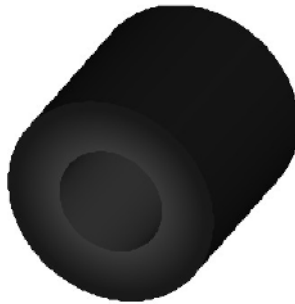
### 8.3.1 FGM pipe

FGM is usually used for multiple objectives, for example in the reduction of thermal stresses and shielding of heat flow in a piping system. The idea is to use heat resistance ceramics on the high temperature side and high strength metal with high thermal conductivity on the low temperature side, with a gradual material variation from one material phase to another. The outcome is the preparation of a new composite with a gradual compositional variation from heat resistant ceramics to fracture resistant metals. The determination of an optimal material distribution of the two materials can be done in order to control the thermoelastic stresses induced by the temperature gradient.

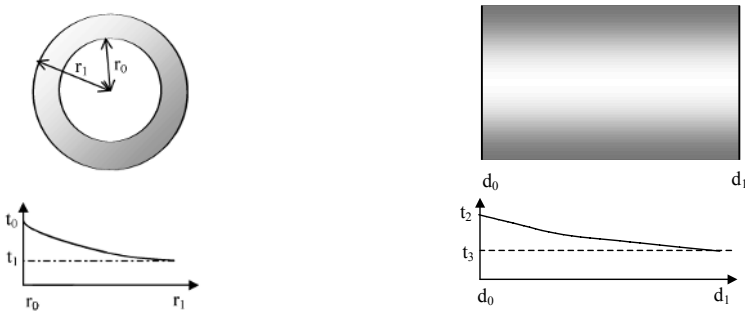
Consider a long pipe for transporting pressurized hot steam, using metal at the outer surface and ceramic at the inner surface to fulfil both the high strength and high temperature resistance requirements respectively. Material grading is required in order to reduce the thermal stress induced by the thermal mismatch of the inner and outer surfaces of the pipe. An equation ( $f(d) = a_0 + a_1x + a_2x_2$ ) proposed by Markworth and Saunders (6) is used for governing the material variation, as shown in Fig. 8.5.

For a complicated heat conduction piping problem, the temperature not only changes from the inner surface to the outer surface, but also across the pipe as shown in Fig. 8.6.

For this case, the material composition may be varied in two directions, i.e. from the inner surface to the outer surface and from one end of the pipe to the other end. In the modelling system, this can be



**Fig. 8.5 A pipe using FGM**



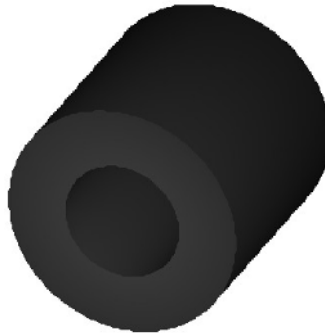
**Fig. 8.6 (a) Heat profile across cross-section of the pipe.  
(b) Heat profile along the pipe**

done by using two different grading sources unioned by the ‘ $\otimes$ ’ operator described earlier. Figure 8.7 depicts an example of this case.

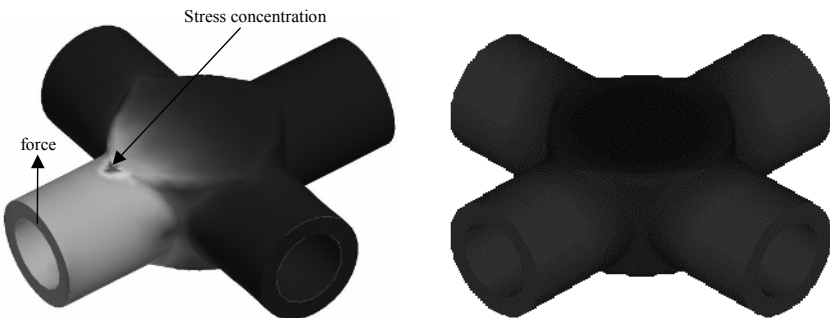
In addition to the above, consider a model of a cross-tee assembly in a piping system, when there is a force acting on one of the pipe ends of the assembly, the stress concentration occurs near the edges of the joint as shown in Fig. 8.8a. This phenomenon can be suppressed by introducing stronger materials in the form of an FGM at the stress concentration area as in Fig. 8.8b.

### 8.3.2 Watch assembly

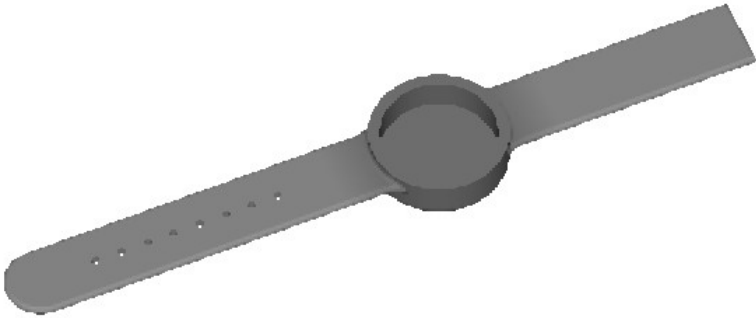
FGM can also be used in joining different materials to reduce interfacial stresses between materials. Figure 8.9 shows a combination of a watchcase and watchband. The material used in the watchcase is



**Fig. 8.7 A pipe with material gradings in two directions**



**Fig. 8.8 (a) Cross-tee assembly under a shear stress. (b) Cross-tee assembly has enhanced material at the stress concentration area**



**Fig. 8.9 Elastic watchband and high strength watch case produced using FGM**

a high strength material and that for the watchband is of another type. By using FGM, these two materials can be joined together and produced in one go without the need for a pin joint.

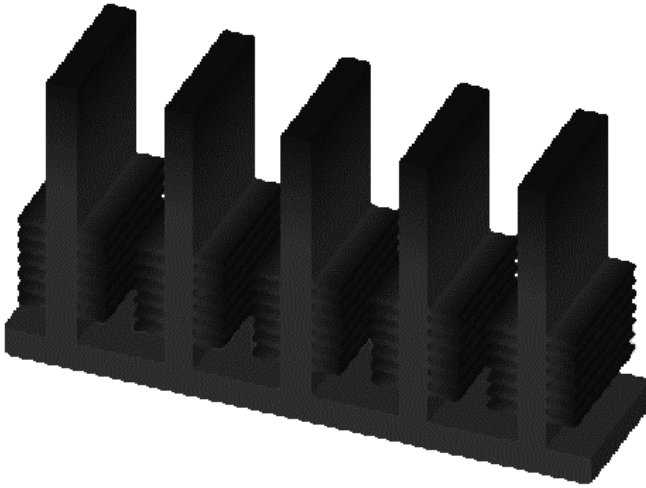
The use of FGM in joining materials shows a new way of simplifying the design and assembly process of the product, with a view to reducing the total production cost.

### **8.3.3 Cooling fins**

In electromechanical and electronic components such as the CPU of a computer, the ability to maintain the operating temperature within an acceptable range is crucial to the normal operation of the device. A natural convection cooling fin is a commonly used feature to increase the heat dissipation rate.

In Fig. 8.10, it can be observed that there is a material grading in each main-fin and sub-fin of the whole cooling fin. This can be done by applying different plane grading sources on each material grading position.

By using a heterogeneous solid modelling CAD system, we can optimize the design of the cooling fins. The heat conductivity along the fins can be modelled according to the heat conduction profile by controlling the materials variation in order to increase the cooling efficiency. The thermal stresses within the fins can also be minimized.



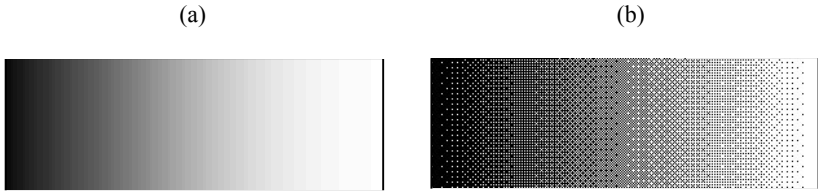
**Fig. 8.10 FGM in a multi-fin array configuration**

#### **8.4 SLICING AND CONTOURS GENERATION FOR FABRICATING HETEROGENEOUS OBJECTS**

Jackson (7) developed a three-dimensional printer for fabricating heterogeneous objects. The working principle of this three-dimensional printer is similar to the common ink-jet printers, i.e. the printer head prints different materials in each slice in a drop-by-drop manner. Similar to ink-jet printing, three-dimensional printing is a binary device. During the printing process, it can only decide either 'to print' or 'not to print', i.e. a 'black and white' bitmap. The continuous material grading is converted into a discrete version of machine instructions by a technique of digital half-toning as shown in Fig. 8.11.

The digital half-tone printing suffers from the disadvantages of requiring enormous storage space and less-than-desirable precision. This is because it is difficult to fabricate heterogeneous objects with material distributions varying from point to point like printing a bitmap.

Apart from the 'drop-by-drop' method of Jackson, another widely used method includes a variety of powder layering techniques (8). Under these techniques, the continuous material grading is discretized into step-wise gradings. Materials are mixed into a desired volume ratio and built step-by-step according to the material compositions.



**Fig. 8.11 (a) Original material grading. (b) Half-toned grading**

For the reasons mentioned earlier, the digital half-toning method, however, is not applicable to the powder layering techniques. A more general layered representation is thus necessary. Apart from storing each layer as an array of pixels or voxels, as in reference (9), some other common sliced formats can also be used. These commonly used file formats consist of successive cross-sections taken at ascending  $Z$  intervals in which solid material is represented by interior and exterior boundary polylines (contours) as in the CLI (common layer interface) and SLC (stereolithography contour) file formats.

Although these file formats are unable to represent material information inside the contours, they can be modified and algorithms for generating contours which support material information can be established. The basic idea is to subdivide each slice of a heterogeneous object with functionally graded materials into a list of multi-material contours according to the grading functions  $f(d)$  of different grading sources.

#### 8.4.1 Contours generation

As mentioned in Section 2, three regions compose a grading source. Because the material compositions inside the +ve and -ve complementary regions are constant throughout, contour subdivision is only necessary in the effective grading region. To subdivide a continuous material grading into discrete contours of material grading, a parameter called 'grading step-width'  $t_g$  is defined ( $0 < t_g \leq 1$ ) to control the resolution of the grading. With this grading truncation and contour subdivision, a heterogeneous object can be built by layered manufacturing machines of different fabricating precision specification since the degree of precision of the material grading during fabrication can be controlled.

The number of contours that can be generated within the effective grading regions depends on  $t_g$ . The smaller the step-width, the larger is the number of the contours generated.

The number of contours in the effective grading region is given by

$$N_c = \frac{1}{t_g} + 1$$

The  $f(d)$  intensity of the different contours during subdivision is defined by the following equations:

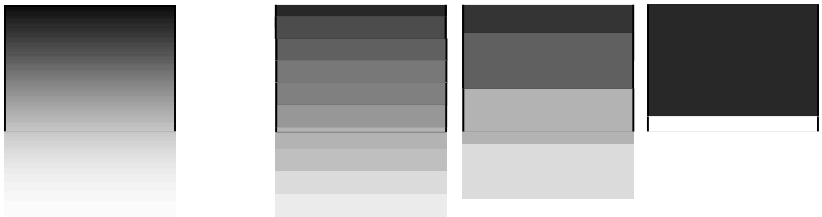
$$f(d) = 0 \quad \text{for } 0 \leq f(d) < \frac{t_g}{2}$$

$$f(d) = t_g \times (u + 1) \quad \text{for } \left(\frac{1}{2} + u\right) \times t_g \leq f(d) < \left(\frac{1}{2} + u + 1\right) \times t_g \text{ and} \\ u = 0 \dots N_c - 3, u \geq 0$$

$$f(d) = 1 \quad \text{for } \left(\frac{1}{2} + N_c - 2\right) \times t_g \leq f(d) \leq 1$$

The material composition of each step is redefined according to the truncated  $f(d)$  equation. The equations ensure that when the step-width is set to maximum (i.e.  $t_g = 1$ ), the grading can be eventually truncated into a multiple material object with sharp material boundary as shown in Fig. 8.12d.

For example, if  $t_g = 0.25$ ,  $N_c = \frac{1}{0.25} + 1 = 5$  contours are formed.



(a) Original  
grading ( $t_g \ll 1$ )

(b)  $t_g = 0.1$

(c)  $t_g = 0.25$

(d)  $t_g = 1$

**Fig. 8.12 Sliced contours of different step-widths**

$$\begin{aligned}
f(d) &= 0 && \text{for } 0 \leq f(d) < t_g \times 0.5 \\
f(d) &= t_g \times 1 && \text{for } t_g \times 0.5 \leq f(d) < t_g \times 1.5 \\
f(d) &= t_g \times 2 && \text{for } t_g \times 1.5 \leq f(d) < t_g \times 2.5 \\
f(d) &= t_g \times 3 && \text{for } t_g \times 2.5 \leq f(d) < t_g \times 3.5 \\
f(d) &= t_g \times 1 && \text{for } t_g \times 3.5 \leq f(d) < t_g \times 4
\end{aligned}$$

The sub-division of the grading is shown in Fig. 8.12c.

From the above example, it is seen that by varying the step-width, the material grading can be altered from a simple multiple material case with sharp material boundary ( $t_g = 1$ ) to one with smooth grading ( $t_g \ll 1$ ).

A model shown in Fig. 8.13a is used to demonstrate the contour subdivision algorithm of a slice. First, each slice is formed by sampling the material composition of a lattice of points within the geometric boundaries of each layer of the heterogeneous object (Fig. 8.13b). The  $f(d)$  intensity map is thus generated as shown in Fig. 8.13c.

Figure 8.14 shows that the continuous material grading in the effective grading region is truncated into a stepwise grading by specifying the step-width  $t_g = 0.25$ . The number of contours generated in the effective grading region is 5.

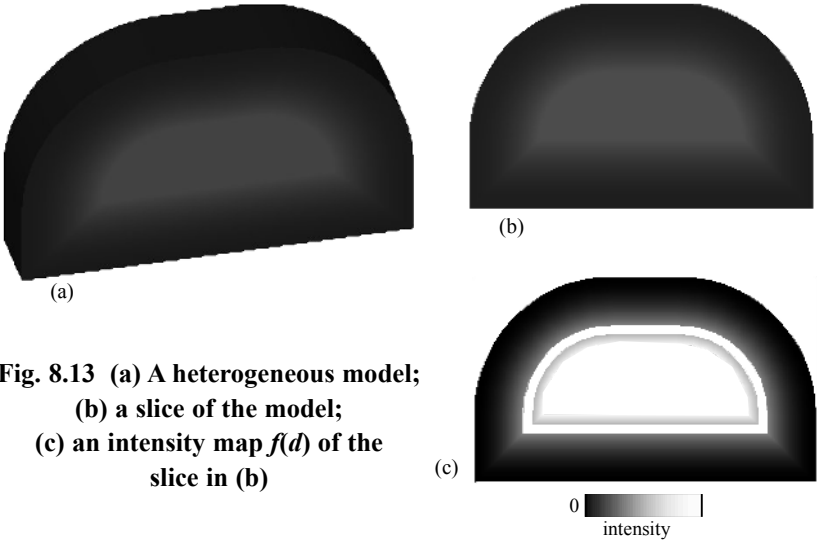
Contours of the three grading regions are then generated according to the grading regions and the intensity of the truncated  $f(d)$ . The total number of generated contours is the summation of the contours generated in these three grading regions. A series of contours is then formed as shown in Fig. 8.15.

As each contour  $C$  has its own material composition array  $M$ , the problem becomes a multiple material object's slice representation which has been elaborated by Chiu and Tan (5). Based on that paper, a tree structure was generated to extract the material composition of the contours in each slice.

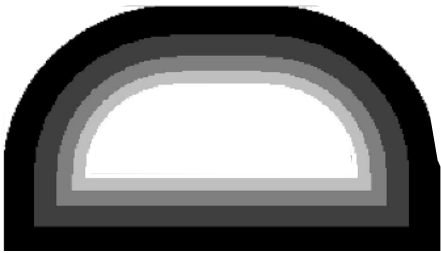
#### 8.4.2 Contours extraction for multiple grading sources

The effect of combining different grading sources with different material grading in a heterogeneous object via an intersecting operator ' $\otimes$ ' was presented in the previous section and the material composition in the intersection regions between grading sources can be calculated.

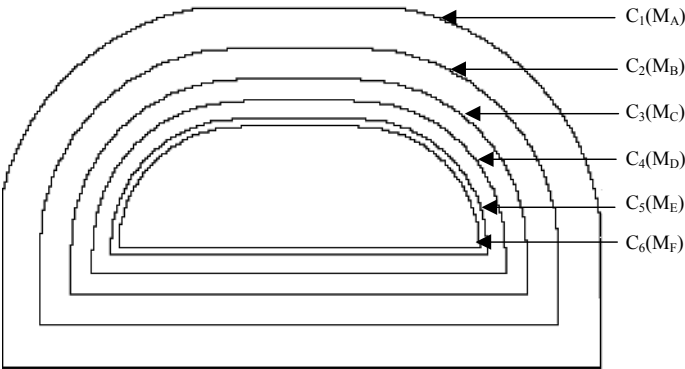




**Fig. 8.13** (a) A heterogeneous model;  
(b) a slice of the model;  
(c) an intensity map  $f(d)$  of the  
slice in (b)



**Fig. 8.14** Truncated stepwise grading



**Fig. 8.15** Total number of contours generated according to the  
truncated  $f(d)$

Upon slicing the object and on each layer, the grading sources are decomposed into stepwise grading. Different step-widths ( $t_g$ ) can be given to different grading sources and contours for the effective grading regions of different grading sources are generated independently as described in Section 8.4.1. Each contour  $C$  in a slice contains a homogeneous material region  $R$  within which the material composition is the same.

$$R = (C(M), I)$$

where

$C \in$  points on the contour of the region with material composition array  $M$

$I \in$  points inside the region

Intersection between grading sources leads to intersection of contours in a layer during the slicing process. In order to ensure that there is no overlapping between contours, those overlapping regions, which are caused by intersection of grading sources, are extracted out. A new contour and the material composition of this region will be generated. For example, if  $C_1(M_A)$  and  $C_2(M_B)$  are contours belonging to grading source  $G_1$  and  $G_2$  respectively, each contour has its own material composition stored in two-material composition arrays  $M_A$  and  $M_B$ .

$$R_1 = (C_1(M_A), I_1), \quad R_2 = (C_2(M_B), I_2)$$

During intersection, three new regions are defined. The material compositions and contours of these regions are regenerated.

Intersecting region  $R_{12} = R_1 \cap^* R_2 = (C_{12}'(M_A \otimes M_B), I_{12}')$

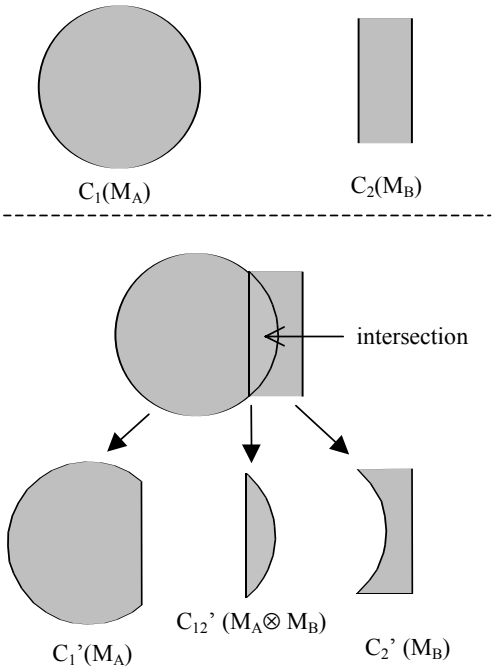
$$R_1' = R_1 / * R_{12} = (C_1'(M_A), I_1')$$

$$R_2' = R_2 / * R_{12} = (C_2'(M_B), I_2')$$

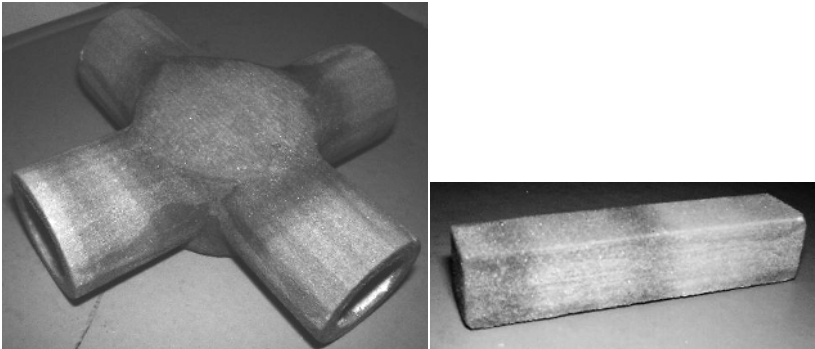
with  $/ *$  and  $\cap^*$  denoting the regularized difference and intersection operators

A schematic diagram of the extraction of the intersecting contour is shown in Fig. 8.16.

Three new subcontours,  $C_{12}'(M_A \otimes M_B)$ ,  $C_1'(M_A)$ , and  $C_2'(M_B)$  are formed and a new tree structure is regenerated according to the new contours list.



**Fig. 8.16 Schematic diagram of the extraction of the intersecting contour**



**Fig. 8.17 Prototypes with colour (material) variation are built by the colour three-dimensional printer**

## 8.5 CONCLUSION

By using the mathematical model derived in this paper, heterogeneous objects with functional variations in materials can be modelled in CAD systems. By using different grading sources, different types of material grading effects can be embedded in a heterogeneous model. The representation scheme and examples of using the proposed modelling scheme are implemented on the Unigraphics V16 CAD system. Two samples using different colours to simulate different materials have been built in order to verify the modelling scheme as shown in Fig. 8.17. In addition, a contour subdivision method by truncating the material grading from continuous grading to stepwise grading for fabrication is also discussed. By adjusting the 'grading step-width'  $t_g$ , the precision for fabrication can be adjusted.

## ACKNOWLEDGEMENTS

The authors would like to thank the Department of Mechanical Engineering, The University of Hong Kong, and the Research Grant Council for supporting this project.

## REFERENCES

- 1 Suresh, S. and Mortensen, A. (1998) Fundamentals of functionally graded materials. *IOM Communications*.
- 2 Kumar, V. and Dutta, D. (1998) An approach to modeling heterogeneous objects. *ASME Journal of Mechanical Design*, **120**(4), December.
- 3 Siu, Y. K. and Tan, S. T. (2001) Source-based heterogeneous solid modeling. *CAD Journal*, **34**(1), January, 41–55.
- 4 Jackson, T. R., Liu, H., Patrikalakis, N. M., Sachs, E. M., and Cima, M. J. (1999) Modeling and designing functionally graded material components for fabrication with local composition control. *Materials and Design*, **20**(2/3), 63–75.
- 5 Chiu, M. J. and Tan, S. T. (2000) Multiple material objects: from CAD representation to data format for rapid prototyping. *CAD*, **32**(12), 707–718.
- 6 Markworth, A. J. and Saunders, J. H. (1995) A model of structure optimization for a functionally graded material. *Materials Letters*, **22**(1–2), January, 103–107.
- 7 Jackson, T. R. (2000) Analysis of functionally graded material

object representation methods. PhD thesis, Massachusetts Institute of Technology, June.

- 8 Xia, M., Hamada, H., and Maekawa, Z. (1995) Flexural stiffness of injection molded glass fiber reinforcement thermoplastics. *International Polymer Processing*, **5**, 74–81.
- 9 Whitney, J. M. and McCullough, R. L. (1989) *Delaware Composites Design Encyclopedia*, Volume 2, *Micromechanical materials modeling* (Technomic Publishing Company).

# Chapter 9

## Reverse engineering and rapid prototyping

**S Remy<sup>1</sup>, A Bernard<sup>2</sup>, G Ris<sup>1</sup>, O Nartz<sup>3</sup>, Y F Zhang<sup>4</sup>, H T Loh<sup>4</sup>,  
and Y S Wong<sup>4</sup>,**

<sup>1</sup>Université Henri Poincaré, CRAN, Nancy, France

<sup>2</sup>Ecole Centrale de Nantes, IRCCyN, Nantes, France

<sup>3</sup>Université Henri Poincaré, AIPL, Nancy, France

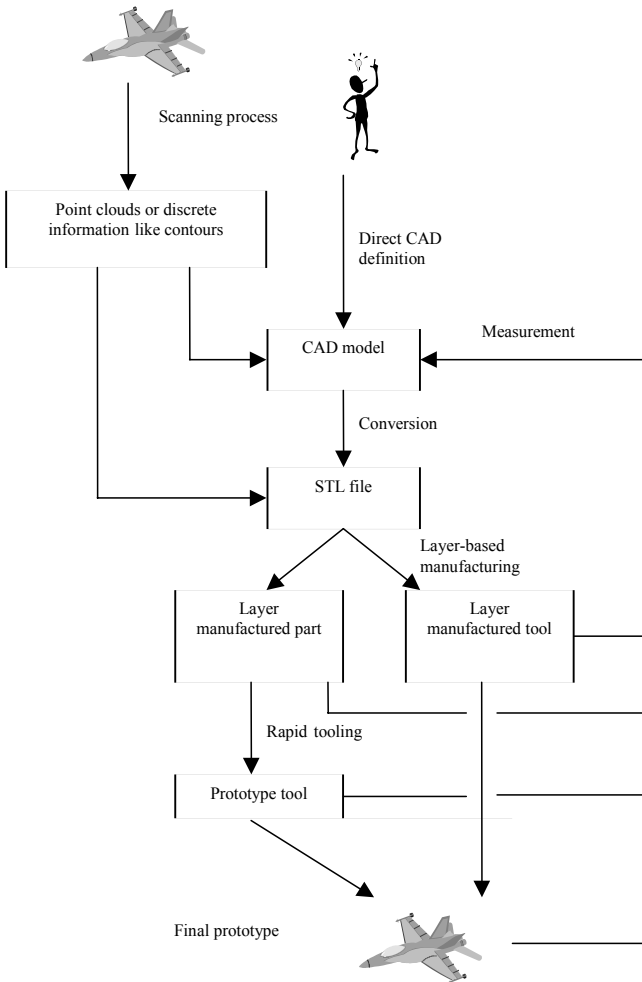
<sup>4</sup>National University of Singapore, Mechanical Engineering  
Department, Singapore

### 9.1 INTRODUCTION

Manufacturing companies are being forced to produce products more and more quickly, at lower costs and with increasing quality in a more competitive market. Rapid prototyping processes (**1**) now make it possible to optimize design time and the manufacture of new products, see Fig. 9.1.

Among the key elements of this process, digitization (acquisition and modelling of forms) converts a physical object to its virtual representation. Contact sensors are widely used for measuring mechanical parts in industry. For a part with complex shape, however, non-contact sensors are more suitable. Digitization allows the control of parts, comparison between the result of digitization and the CAD reference model, and measurement of precise geometrical characteristics. But digitization with a non-contact sensor remains a complex process to implement.

There are many CAD applications which use sensors. Generally, these applications can be categorized into two types (2). The first is for surface reconstruction when we have a part with no CAD model. The digitization process means that less time is needed to obtain a complete CAD model of the part or an STL file, where the surface of the object is represented by triangular facets. The virtual model can be used in some simulation or virtual prototyping environments. We can also use the STL file to produce the piece with processes like SLA. We consider this type of application as reverse engineering. The resultant CAD



**Fig. 9.1 Reverse engineering in rapid prototyping**

model can be modified to improve some characteristics, used for CNC (computer numerical control) machining processes or RP (rapid prototyping) processes.

The other type of application involves the inspection of complex parts. Here, the CAD model and the point cloud from scanning the object are first registered. Then, a map of the error is generated by comparing the CAD model and the point cloud.

In this chapter, we will present, in the first part, a review of all the techniques that are used to obtain discrete data from the surface, the different types of sensor, their characteristics, and the criteria of choice. In the second part, we will present some widely employed reverse engineering approaches, which include the methods to process point clouds, rebuild a surface, and obtain an STL file to make a copy of the object. In the third part, we will discuss several methods for the inspection of the geometry of the made object using three-dimensional digitizing, as well as the existing methods that improve the accuracy of the acquisition.

## **9.2 HOW TO OBTAIN POINT CLOUDS – THREE-DIMENSIONAL DIGITIZING**

First, we present the various sensor technologies available on the market that can be used to digitize an object, as well as a brief outline of how data resulting from these sensors can be best exploited. The digitization of three-dimensional shapes allows the translation of a physical part, see Fig. 9.2, to its digital model, see Fig. 9.3.



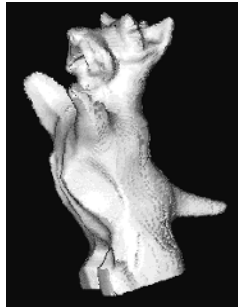
**Fig. 9.2 Physical model**



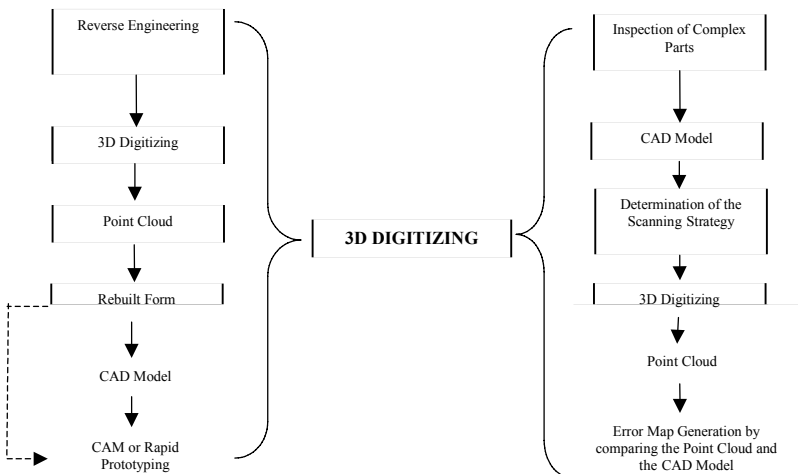
As can be seen from Fig. 9.4, digitization can be used to (3):

- Rebuild the form of an object (physical prototype, model validated in experiments, ergonomic part) in order to obtain its CAD model.
- Compare numerical discrete information (points) delivered by the sensor with the CAD reference model; in such a case, the objective is the control of the part.
- Generate information necessary for the manufacturing process, in order to copy an object (CNC file for machining or STL file for layer-manufacturing).

The numerical data resulting from this means of acquisition generally appear as a point cloud, see Fig. 9.5. A point cloud is an assembly of



**Fig. 9.3 Digital model**



**Fig. 9.4 Three-dimensional digitizing issues**

independent points, each point being known by its Cartesian coordinates in three-dimensional space.

Several papers on state-of-the-art reverse engineering have already been published (4–9) together with several articles on digitization and reverse engineering (10–11). Lists of sensor manufacturers have also been drawn up (7, 9, 12). However, offering a broad outline of the currently available sensors, defined by their technique of data acquisition, seems a useful proposition. Sensor techniques are usually divided into two main categories: contact sensors and non-contact sensors. Further, contact sensors can be subdivided into point-to-point sensors and analogue sensors. Non-contact sensors can be subdivided into optical sensors with a laser, optical sensors without a laser, sensors used in medical imagery, and sensors using the principle of flight time.

Another mode of classification has been proposed (7), where sensors are divided into three categories according to data acquisition mode: (1) passive data acquisition, (2) active data acquisition, and (3) the CGI sensor ('destructive combined mode'). In the following sections, these three types of sensors are introduced with a discussion of their characteristics, possible support, point cloud data obtained, and the vendors.

### 9.2.1 CGI process

CGI company (capture geometry inside) {website 1} offers the CSS-1000 system. It is based on a technique, which alternates a machining of the part then a digitization of each layer, using an optical scanner. This principle has the advantage of allowing the digitization of the internal geometry of the part. The disadvantage is that the part has to be destroyed.



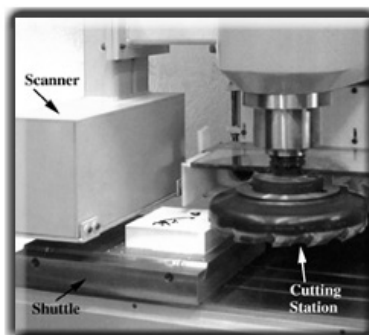
**Fig. 9.5 Point cloud**

The first stage of this process is preparation of the part. It is necessary to completely embed the part in a resin, which makes it possible to maintain the integrity of the part during the destruction phase. Its colour is selected in order to accentuate contrast between the part and the resin at the time of digitization. The resin must thus fill all the cavities of the part and must completely include it in order to form a rectangular compact block. The machine is able to machine a block of maximum size  $300 \times 260 \times 200$  mm. During the second stage, the user must inform the system of various parameters (size of block, type of material, etc.). The remaining parameters (spindle speed, feed speed, depth of cut, etc.) are then automatically defined by the system. Then, the system alternately machines the block, see Fig. 9.6, and scans it.

In the machining–scanning process, a mill is used to remove one layer of a thickness of 0.012–0.25 mm. Once the layer is completely removed, an optical scanner acquires the image of the section. This operation is repeated until the block is completely machined. When the part is completely scanned, the whole image must be computed. In the final stage of the process the contours of the part, interior and exterior, are created. Each one of these contours is composed of a succession of points, see Fig. 9.7.

### 9.2.2 Passive acquisition mode

Acquisition is known as passive when the sensors do not exert any influence on the part to be digitized, or on its environment. A camera is used, which provides one or more images. The information of depth must then be extracted by the analysis from each image, for example, by processing the levels of grey in the image.



**Fig. 9.6** CGI process ([www.reverse-eng.com](http://www.reverse-eng.com))



**Fig. 9.7 Resulting point cloud ([www.reverse-eng.com](http://www.reverse-eng.com){1})**

One of the passive acquisition methods is based on focus and perfect knowledge of the elements of definition (i.e. focal distance and focal plane) of the optical system, which makes it possible to calculate the distance between a point and the sensor. Despite offering more precise measurements than the active method of triangulation by laser, this technique remains largely unused because of the optical adjustments to be carried out by the operator.

Photogrammetry (13) is probably the best-known passive acquisition method. It is also sometimes called stereovision. It defines the shape and the position of objects, reconstitutes them, and determines some of their characteristics starting from several photographs of the same object, corresponding to the different points of view. Photogrammetry, until now, has been primarily used for the production of topographic cards, starting from air photographs.

There are also applications in the automotive design field, production, and assembly control (14). The sensors used are in fact special cameras called chambers. These sensors are often numeric in order to obtain an immediate result of the photos. Two types of sensors exist: independent photogrammetric chambers and stereometric chambers. In fact, the stereometric chambers are used for close-up photogrammetry. An independent photogrammetric chamber is, as its name indicates, self-contained equipment. This type of sensor is used when it is advisable to adapt the distance from catch of sights and the length of the base in order to obtain high precision details.

The process itself applies itself particularly well to use with naturally textured materials (wood, textile, paperboard), but can also be applied

to non-textured materials (plastic, metal) if lighting conditions are adapted. Stereovision also enables automatic location of an object in a three-dimensional environment by known complex pattern recognition. Orteu (15) determines, by stereovision, the deformation (evolution of form between two states) of a sheet metal part by projecting a regular grid on the part before forming and by analysing the pair of photographs of the initial state and the final state. Ip and Loftus (16), thanks to the use of an opposite illumination algorithm, determine the normals on the surfaces of a statue starting from several numerical photographs. Surfaces are then rebuilt and a program of machining for a three-axis milling machine with numerical control is generated in order to reproduce the part. Cadio (14) uses dense stereovision, starting from images obtained using a stereomicroscope or two cameras, for the topography of the surface of objects.

### 9.2.3 Active acquisition mode

Contrary to the passive acquisition mode, the sensors classified in this mode interact with the part. This interaction can be carried out in various ways:

- by physical contact between the sensor and the part;
- by emission of light, generally structured, then acquisition by an optical system based on the luminous trace left on the part;
- by the use of electromagnetic waves.

#### 9.2.3.1 *The contact sensors*

This is the most widespread mode of data acquisition. The contact sensors are usually used in industry on three-dimensional co-ordinate measuring machines (CMM). The sensors generally consist of a sensor equipped with a touch probe, made up of a stem finished by a spherical end, which delivers a signal as soon as pressure is exerted on the probe. However, it is necessary, because of the principle of acquisition, to carry out a correction of the radius of the probe to obtain real measurement. This correction is generally carried out in an automatic way.

Despite their precision, the contact sensors have some disadvantages:

- They are relatively slow: the speed of acquisition is approximately 10 points per minute in manual mode, and can reach approximately 100 points per minute in automatic mode.

- They are unable to acquire details lower than the size of the probe.
- They cannot acquire correct information on soft material parts: the part is deformed when the probe comes into contact with the surface, thus distorting measurement.

As stated previously, contact sensors can be divided into two categories:

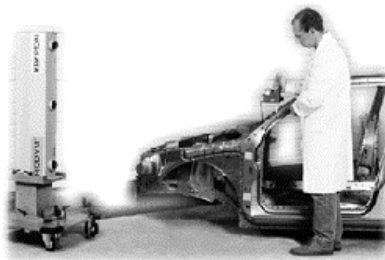
- point-to-point sensors, which allow specific acquisitions;
- analogue sensors, which follow contours or surfaces automatically.

The point-to-point sensors, see Fig. 9.8, are, in the majority of cases, installed on a CMM. They can also be assembled on an articulated arm.

Another point-to-point probe technology is based on camera observation of a pointer or target, see Fig. 9.9, positioned on the object, which works according to the principle of triangulation. The system containing targets is dedicated to the acquisition of the movement of an object. A limitation of these sensors is that there can be no obstacle between the probe and the receiver.



**Fig. 9.8 Point-to-point sensor ([www.romer.fr](http://www.romer.fr){2})**



**Fig. 9.9 Target probe and cameras ([www.krypton.be](http://www.krypton.be){3})**



**Fig. 9.10 Analogue sensor (www.renishaw.com{4})**

The analogue sensors, see Fig. 9.10, use a similar technology to the point-to-point sensors. Thanks to an adapted sensor, it is possible to follow contours or surfaces automatically. In the case of surfaces, digitization is generally carried out according to a squaring of the part. Although more effective than the point-to-point sensor, they have the disadvantages of reduced speed, accessibility, and reliability on some tender or soft material parts.

#### **9.2.3.2 Non-contact sensors**

Non-contact sensors can overcome the problems associated with contact sensors. Indeed, the data acquisition on the surface of parts does not require physical contact between the sensor and the part. Thus, parts do not undergo any deformation. The non-contact sensors can also be subdivided into several categories:

- based on time of flight;
- resulting from medical imagery;
- based on optical techniques without a laser;
- based on optical techniques with a laser.

##### *Principle of acquisition per measurement of the time of flight*

These sensors use the principle of radar. They emit a wave, of which the propagation velocity is known, and measure the time run out between the emission and the return of the wave reflected on the surface of an object, either by the object itself or by a target. Consequently, it is possible to determine the distance between the sensor and the point aimed on the object. These sensors are sometimes

also called telemeters. The advantage of these sensors is that they are able to digitize distant and large-sized objects. The principal disadvantage is that the measurement taken by these sensors is specific; a sweeping of an object (or environment) is thus necessary in order to obtain the totality of information, thus slowing down the process of digitization. This type of sensor is also limited to the data acquisition of non-metal objects. It is also interesting to note the work of Cadio (17), which develops a laser telemeter that makes it possible to operate in an underwater area, in order to improve localization and attitude of a system operating by remote control.

#### *The sensors resulting from medical imagery*

These sensors are generally known by their medical applications, but they are now also used by industry. This acquisition method can digitize the internal part or deep cavities of a part, which are generally inaccessible with the other acquisition methods, except for the sensor proposed by CGI. The principal disadvantage is the high cost of purchase. The principles used by these sensors are the following:

- the echography that uses ultrasound for generating the cut plane of the digitized part;
- the NMR (nuclear magnetic resonance), which uses a magnetic field for obtaining a detailed cut of the part;
- the tomography by X-ray.

Tomography measures the quantity of X-rays not absorbed by the part in order to provide a geometrical cut, including density information. Dastarac (18) used the Magic tomograph of Intercontrôle to digitize a pump housing for the turbine of a helicopter for which there was a correct CAD model, in order to validate the tomograph-like tool of retrodesign. He also digitized the head cylinder of a fuel injection engine in order to create a CAD definition. Ménégazzi *et al.* (19) used a medical scanner from General Electric to digitize the head cylinder of an engine, in order to extract the geometry of the cooling system to analyse it thereafter. Rollins and Jurick (20, 21) used a tomograph of Scientific Measurement System {website 5} to digitize the exhaust manifold of a truck engine, carried out in a foundry, in order to compare the critical section with the CAD model and to validate the tool. This sensor was also used to digitize another collector, in order to carry out a finite element analysis on a model with a grid of the part, then to rebuild surfaces of the part in order to obtain a CAD model. Neel and Yancey (22) used the tomography to rebuild the moulding

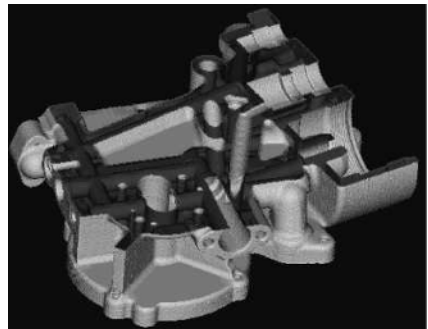


cores of a gearbox. All these sensors generate sections of the object to be digitized. It is the assembly of all the parallel sections of the object, which makes it possible to reconstruct the geometry of the part. The Velocity software, from Image3 {website 6}, and Mimics, published by Materialise {website 7}, allow computing of data of nuclear magnetic resonance sensor or tomography in order to obtain CAD models. Griffin *et al* (23) used Velocity to rebuild an STL model of a bottle of water and a fossil to create a copy by rapid prototyping. Tomoadour, in France, is a company that uses tomography, see Fig. 9.11 {website 8}.

Nuclear magnetic resonance (NMR) {website 9} is defined by Joseph P. Hornack as a phenomenon which occurs when the nuclei of certain atoms are immersed in a static magnetic field and exposed to a second oscillating magnetic field. Some nuclei experience this phenomenon, and others do not, dependent upon whether they possess a property called spin. NMR can be used to create an image. Volume imaging is the acquisition of magnetic resonance data from a volume rather than a tomographic slice. It can be thought of as collecting several contiguous slices through a region of an imaged object. The number of contiguous slices will always be a multiple of 2.

#### *Optical techniques without a laser*

Here, there are two relatively similar principles: (1) sensors with structured light and (2) sensors for the purpose of creating a Moiré effect. The two techniques project on the part a regular optical grid, a network of fringes, and calculate the co-ordinates by triangulation. They differ in the manner in which they observe the scene: the first is

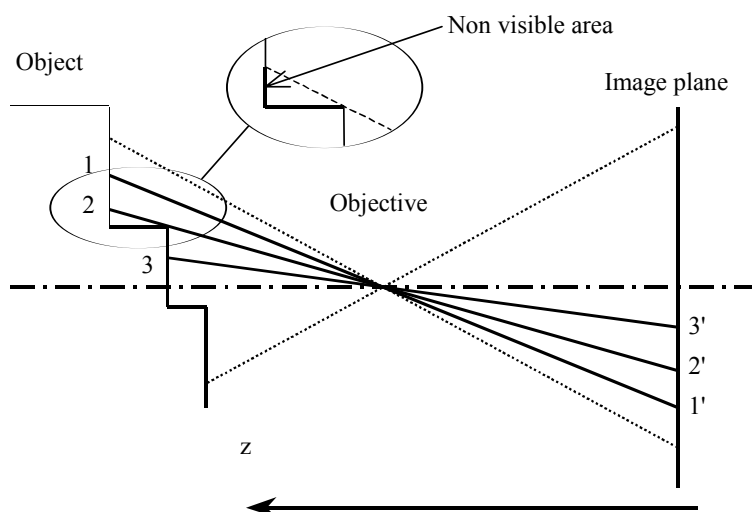


**Fig. 9.11 Tomography (tomoadour.fr{8})**

simply satisfied to observe it, while the second observes the scene through the same regular optical grid, but with orientation different from that projected on the part. A limit of these sensors is the aliasing-type surface, see Fig. 9.12. It is then necessary to carry out other acquisitions, with different orientations.

A series of networks of fringes is projected on the object and the camera acquires each image. The interpretation of these images then makes it possible to deduce the three-dimensional co-ordinates from the points on the surface of the part.

The Atos sensor functions on this principle. The advantage of this sensor is to combine two cameras, thus making it possible to multiply the number of points acquired with each digitization. Moreover, it is able to digitize objects whose size varies between 10 mm, in only one acquisition, to 5 mm, in several acquisitions (24), by adapting the optical parameters of the cameras after their calibration. Steinbichler {website 10} proposes the Comet sensor in several versions (25, 26). These sensors are preferably installed on a CMM. The Comet/OptoTrak system combines a Comet sensor to digitize the part and an OptoTrak sensor to locate the Comet sensor in the workspace and thus to readjust the point clouds automatically.



**Fig. 9.12 Limit of structured light process**

Let us detail the phenomenon of the Moiré effect. Initially, it projects a network of fringes (projected frame) on the surface of the object. The source of light is generally a halogen lamp. Secondly, the observation of the scene is carried out through the same network of fringes, but directed differently with regard to the first network (frame reference). The superposition of the frames creates the Moiré effect, see Fig. 9.13; interference rings are observable on the two-dimensional image and give information on the relief of the object via level lines. This technique is used to calculate the relief, deformation, and dimensions of a part in a given plan.

#### *Optical techniques with a laser*

This principle of measurement is based on trigonometry, and more particularly on triangulation. The part is lightened by a coherent source of light (a laser beam in general), and a CCD (charged coupled device) camera observes the scene under a different angle. After calibration of the camera, the co-ordinates of a point of the object are deduced starting from the co-ordinates of its image point on the CCD matrix from the camera. The application of matrices of transformation makes it possible to express these co-ordinates in an adequate reference system. The principal disadvantages of these sensors are the phenomena of shade and screening. Indeed, so that a point on the part is acquired, it is necessary that it is enlightened by the source of light and seen by the CCD camera at the same time. If the point is seen by the camera but non-enlightened by the source of light, it is a phenomenon of shade. In the opposite case, when the point is lightened but not seen by the camera, a screening phenomenon is created. The two cameras make it possible to attenuate these phenomena. According



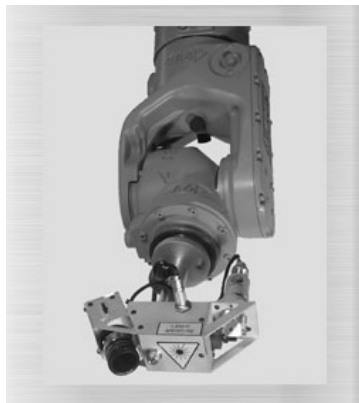
**Fig 9.13 Moiré effect**

to whether the form of the source of light is specific or plane, the sensor is known as laser point or laser plan.

- The laser point, see Fig. 9.14. It is based on the principle of triangulation. A laser diode is used to project a point on the part. The CCD camera that observes the scene thus perceives the image from only one point. The triangulation then makes it possible to extract the co-ordinates from the acquired point.
- The laser plan, see Fig. 9.15. The specific laser beam, provided by a laser diode, is transformed into the laser plan, either by a cylinder lens, or a mirror assembled on a galvanometer (in this case, the movement must be sufficiently fast so that the camera perceives only one trace). The principle of triangulation remains applicable to the image perceived by a CCD camera and also makes it possible to obtain the co-ordinates of a more significant number of points at the same time.



**Fig. 9.14 Laser point (www.digibotics.com{11})**



**Fig 9.15 LR sensor (www.metris.be{12})**

### 9.2.4 Some commercial digitization systems

Based on the sensor techniques introduced in the previous sections, many commercial digitization systems have been developed and made available in today's market. Some of these systems are listed in Table 9.1 (2), together with their embedded techniques and capability specifications.

In summary, many solutions exist to obtain a point cloud of an object's surface. Over the last 10 years, many valid applications have been introduced in the industry and the quality of the acquired data daily improves. However, digitizing systems are only tools for data acquisition; CAD model rebuilding is not an automated process and needs many competencies.

## 9.3 FROM THE POINT CLOUD TO THE OBJECT – REVERSE ENGINEERING

Reverse engineering (RE) refers to creating a CAD model from an existing physical object, which can be utilized as a design tool for producing a duplicate of an object, extracting the design concept of an existing model, or re-engineering an existing part. In other words, RE takes the point cloud of an object's surface captured using the aforementioned digitization techniques as an input and creates a geometric model, which should be compatible with the requirements for a rapid prototyping system or CAM. Since the cloud data are generally dense and unorganized, reconstructing a geometry model for efficient and accurate prototype fabrication becomes a major research issue.

In general, approaches to modelling cloud data can be classified in two categories: (a) surface reconstruction using an implicit function (e.g. parametric function) (27) or (b) surface modelling employing a polyhedral mesh (e.g. triangular mesh) (28). The segment-and-fit approach described by Hoffman and Jain (29) is widely used in the former method. Typically, the cloud data are segmented into several patches bounded by clearly defined curves, each representing a discrete surface region present in the physical object. Modelling methods, such as those employing parametric (30) or quadric (31, 32) functions are then applied to fit surfaces to the patches. Among the parametric representations for curves and surfaces, non-uniform

**Table 9.1 A review of the sensor market**

Vendor	Product	Sensor only	CMM CNC	Articulated arm	Integrates machine	Point-to-point contact	Analogue contact	Non-contact sensor	Maximal volume that can be measured	Accuracy (mm)	Number of points per acquisition (or per second)
3D Scanners	Model Maker	O		O				LL		0.3	14000 pts/s
	Replica	O	O					LL		0.02	14000 pts/s
Bertin Technologies	CTB							TO	30°x30°	1	50000 pts/s
Breuckmann	Optocam							LS	480x300x360	0.02 to 0.1	393216 pts/a
Brown & Sharp	Chameleon				O	O	O	LP	3000x1500x1000	0.0025	180 pts/s
	Scirocco				O	O	O		2600x1200x950	0.0025	500 pts/s
CGI	CSS-100			O					300x260x200		
Digibotics	Digibot II							LP	Ø457x457	0.05	1200 pts/s
EOIS	The Handy	O	O	O				LS		0.075	200000 pts/s
FARO	Faro Arm	O			O		O	LL	Ø3700	0.025	
Geodetic services	V-Star + Inca	O						P	100 to 10000 mm <sup>3</sup>	A1/100000	5000 pts/a
Gom	Tritop				O			P	<20 m <sup>3</sup>	0.01	
	Atos				O			LL	<10 m <sup>3</sup>	0.04	1331300 pts/s
Holo3	Digiplus	O						LS			360000 pts/min
Hymarc	Hyscan 45C	O	O					LL		0.05	10000 pts/s
Immersion	Microscribe	O			O	O	O		Ø1670	0.3	1000 pts/s
Kreon Industrie	KLS 51	O	O	O	O			LL		0.05	17x600 pts/s
	KLS 171	O	O	O	O			LL		0.08	17x600 pts/s
Laser Design	Surveyor 3200				O			LL	750x750x750	0.001	4000 pts/s
Lemoine	Palpage	O	O				O				100 pts/s
Merni	C3D	O						P	Infinite	0.4 to 0.05	500000 pts/a
Mensi	Soisic				O			LP	64000 m <sup>3</sup>	0.3	100 pts/s
Metris	LR 20/100	O		O				LL		0.25	
	LC 20/100	O	O					LL		< 20 µm	
Minolta	Vi-700				O			LL	1100x1100	0.3	40000 pts/a
Mitutoyo	Euro-C Apex				O	O	O		1200x3000x1000	3+4L/1000	
Perceptron	ScanWorks	O	O	O				LL		0.03	20x700 pts/s
Polhemus	Fastscan	O						LL	760x760x760	1	50 pts/s
	3Draw	O				E			450x320x300	0.2	
Renishaw	Cyclone				O		O	LP	600x500x400	0.05	1000 pts/s
Roland	Piczza Pix 30				O			U	150x300x60	0.01	
SAC	SP12				O	U	U		1 to 3 m <sup>3</sup>	0.6	
	MMTP				O	O	O		Ø1200	0.4	30 pts/s
Steinbichler	Comet 500	O			O			LS	460x360x460	0.4	430000 pts/a
	Tricolite	O			O			LS			431000 pts/a
Steinbichler/Zeiss	Autoscan	O		O				LL		0.025	400 pts/s
Stil	CHR 150	O	O					LB	80 to 750 µm	0.0001	1000 pts/s
	Inspect	O	O	O				LB	25x30	0.03	4x560 pts/s
Tricorder	Tricorder				O			P	300x250x250	0.5	4x250000 pts/a
LP : Point Laser LL : Line Laser LS : Structured Light P : Photogrammetry		TO : Optical Telemetry LB : White Light				E : Electromagnetic U : Ultrasound					

rational B-spline (NURBS) is the most popular one due to its ability to approximate accurately most of the surface entities encountered in design and manufacturing applications (33). The model with these kinds of mathematically described surface patches can be used for machining directly. However, segmentation of very large sets of cloud data (manual or automatic) could be a difficult and tedious task. Recently, it is noticeable that some commercial reverse engineering packages combine the polyhedral mesh and parametric surface reconstruction. A typical example is Paraform (34) in which the point cloud is first triangulated followed by a curvature-based mapping method to extract feature curves for segmentation. Parametric surfaces are then created using the feature curves. Therefore, polyhedral mesh can also be used as an intermediate model for final surface creation.

For the approach employing polyhedral meshes, the inherent data structure produced by the vision system plays a critical role in the meshing techniques. The structure in the data can range from being highly organized, such as an array of points, to little structure, such as cloud data. For a highly structured data set, such as a range image composed of a regular grid of data points, a polygonal model can be created in a straightforward manner by linking data points in a neighbourhood to form the mesh. If an object is digitized through the acquisition of multiple range images, then an appropriate registration and alignment technique must be implemented to merge the set of adjoining polygonal domains (28, 35). Generally, algorithms dependent on existing data structure typically perform with greater efficiency than un-constrained algorithms employing unstructured data. However, a major disadvantage of these algorithms is the inherent dependence on specific sensor types, or even manufacturers.

Algorithms that have been developed for modelling less structured three-dimensional data sets assume that no *a priori* information regarding the connectivity of points in the data set is available. The only assumption is that there exists a sufficiently high data sampling resolution to permit unambiguous reconstruction of the model. For example, Fang and Piegl (36) extended a two-dimensional Delaunay triangulation algorithm to three-dimensional data. Cignoni *et al.* (37) described another Delaunay triangulation technique based on a divide-and-conquer paradigm. Lawson (38) used geometric reasoning to construct a triangular facet mesh, and subsequently, Choi *et al.* (39)

extended the same method using a vector angle order, instead of Euclidean distance, to determine the linkage of data points. Recently, Hoppe *et al.* (40) developed a signed distance function by estimating the local tangential plane and using a marching cube method to extract a triangular polyhedral mesh. In this section, three different modelling methods are introduced. The first one utilizes a commercial modelling software program to generate a CAD model from the digitized data. The second one employs an error-based triangulation method to generate a triangular mesh model, while the third one uses a point-based segmentation method to generate a model for RP processes. It is worth mentioning that the last two methods assume that no *a priori* information regarding the connectivity of points in the data set is available.

### 9.3.1 Surface reconstruction using parametric functions

In this case, a radiator cap, see Fig. 9.16 is required to be produced through reverse engineering. The part has two kinds of surfaces: an upper part with a very important aesthetic aspect and a lower part with a functional aspect. Apparently, different approaches may be needed.

#### 9.3.1.1 The method

The whole process involves part digitization, surface rebuilding, fabrication, and verification as shown in Fig. 9.17.



Fig. 9.16 The radiator cap



*From the object to the model*

For the lower and functional part, we choose to measure it with traditional measurement tools like a calliper-square to obtain the dimensions of the different geometrical features (cylinder etc.). To obtain a good numerical image of the upper part, we choose to digitize it. The sensor we use is a laser plane sensor, KLS 51 from Kreon Industrie, which is integrated on a gantry-type of CMM DEA Gamma 1203.

Once the part is digitized, the surface of the upper part of the radiator cap is rebuilt using the SDRC Surfacer software environment. Then, we add on this rebuild surfaces, the lower part of the cap in a CAD environment. In addition to a three-dimensional visualization and an IGES (necessary for CAM) format saving of the complete part, the CAD software permits us to obtain two-dimensional drawings for storage of the lower part characteristics.

*From the model to the object*

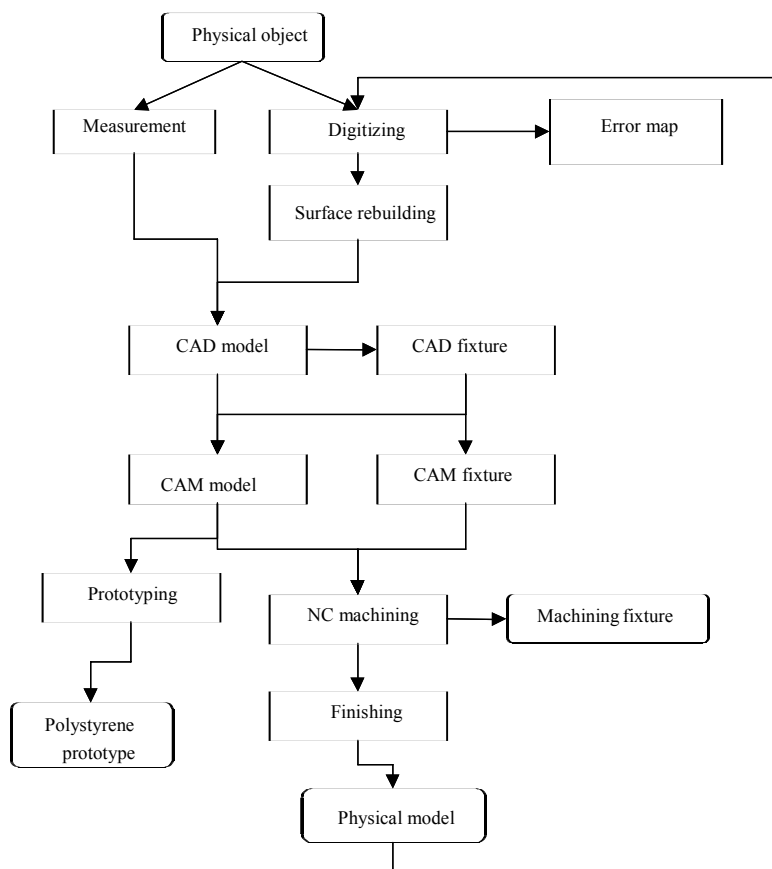
The IGES file is directly imported to a CAM software environment to generate the manufacture strategy permitting the machining of the lower and upper parts. The software automatically generates the ISO codes according to this strategy.

We choose to realize a first polystyrene prototype with a little three-axis milling machine from CharlyRobot to indicate if there are any errors. Once corrections are applied, the new programs are downloaded in a large four-axis milling machine (CU60) for the first piece in a aluminium-based alloy. Then, when the machining is finished, the piece is polished. This task is at present manual, but it will be automated with a six-axis robot (Fanuc S10).

**9.3.1.2 Rebuilding methodology***Three-dimensional digitizing*

From our point of view, the three-dimensional digitizing strategy consists of:

- Finding a position that permits the digitization of all the surfaces of the piece we want to digitize.
- Choosing a density for the point cloud according to the work we want to do (rapid prototyping, class-A surfaces rebuilding, visualization, etc.).

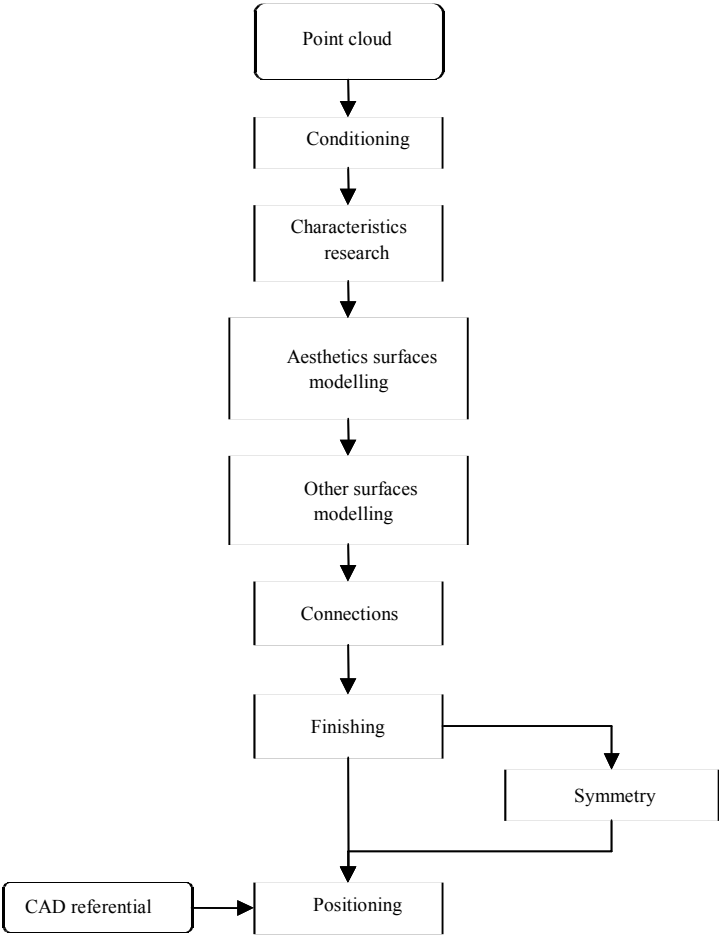


**Fig. 9.17 The global method**

In our case, this strategy is very simple due to the shape we study. We just put the piece flat on the CMM marble, orienting it to put into correspondence the longitudinal axis of the piece and one of the axes of the CMM reference frame. Because of the shape of the piece and because of our machine's capabilities (in terms of the number of degrees of freedom), the whole of the upper part of the radiator cap can be scanned very quickly.

### *Surface rebuilding*

Today, surface rebuilding software (see Fig. 9.18) provides many tools that permit point cloud, curve, and surface treatments but users are free to choose the method they want. Actually, the choice of the method is



**Fig. 9.18 Surface rebuilding methodology**

made according to the objective of the surface rebuilding task. Four families exist with different constraints: CAD, rapid prototyping, visualization, and animation.

Our interest is just in the first case because our aim is to input curves and surfaces in CAD software; for example, from a physical object (a designer’s mock-up) that we scan to reverse engineer it. It is important to try and achieve very good continuity between the surfaces while maintaining surfaces with a great level of accuracy. Then, the model we make will have to be integrated in a CAD environment; there

are, therefore, some restrictions due to the parameters that the software accepts.

After the digitizing operation, we generally have too many points. Due to this, the operations that we make on the point cloud (rotate etc.) are very slow. Then, all the co-ordinates of these points are in the CMM reference frame. The conditioning operation consists of:

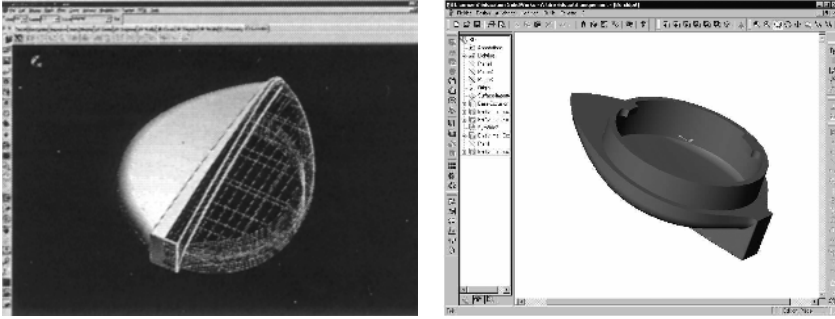
- Sampling the point cloud
- Filtering the point cloud
- Positioning the point cloud in a reference frame
- Searching symmetries if they exist

The characteristics research operation consists of 'reading' the point cloud to allow the scheduling of the different tasks needed. It also permits us to 'understand' how the part was made in the past (type of tools, operations, etc.). This task needs the original physical model. We search with the toolbox of the software all the surfaces that are plane, vertical, or horizontal, the value of the fillets, the cylinders, the sphere, etc. After this work, we cut the point cloud to wrap surfaces on all these characteristic features.

The next stage consists of rebuilding the aestheticism of the part with methods used in car design, like class-A surfaces. The quality of such a model can be measured by using tools permitting the visualization of the reflection of an incident light that has various positions on the created surfaces. During the creation of these surfaces, the point cloud is used only to verify the divergence because the objective of this operation is to obtain the lowest degree surfaces we can, see Fig. 9.19.

The other surfaces are created more classically by using different methods such as wrapping, curve extrusion, square with four curves, etc. They are connected with aesthetic surfaces with respect to position, tangency, and continuity (C0, C1, or C2) criteria. Then, we add fillets between the surfaces as determined in the characteristics research task. At this moment, there are some holes that we must block up with square surfaces defined by four curves.

In the case of a symmetrical part, surfaces that cut the plane of symmetry are cut just before this one. So, after the mirror operation, there is a hole between the two symmetric parts. To block up this hole, we join the two parts with intermediate surfaces with a C1 or C2 continuity.



**Fig. 9.19 Numerical model**

As the rebuilding is completed, we have to remember that the original point cloud was placed in an arbitrary reference frame. In the case of a multi-part product, the CAD model has to be integrated in a predefined CAD environment. So we have to replace our model in its final position. Nowadays, CAD software allows us to realize this task by importing some geometrical references in a neutral format (IGES in our case) that is necessary to integrate the model in its environment, see Fig. 9.19, left.

Then, conventional CAD software (in our case, Solidworks 2000 from Dassault Systèmes) is used to create mechanical and functional shapes, see Fig. 9.19, right.

### **9.3.1.3 Machining**

#### *Rapid prototyping*

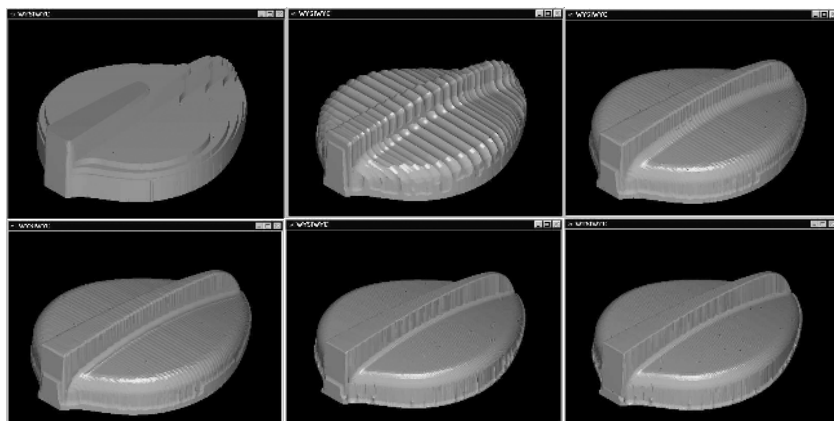
For early detection of any geometrical, aesthetic, or technical problems, we create an initial physical model. As we do not have a layer-manufacturing machine, we make this prototype in a soft material with a small three-axis milling machine. A few minutes later, we have an object that can be manipulated. From this object, we can establish a list of CAD modifications.

#### *CAM*

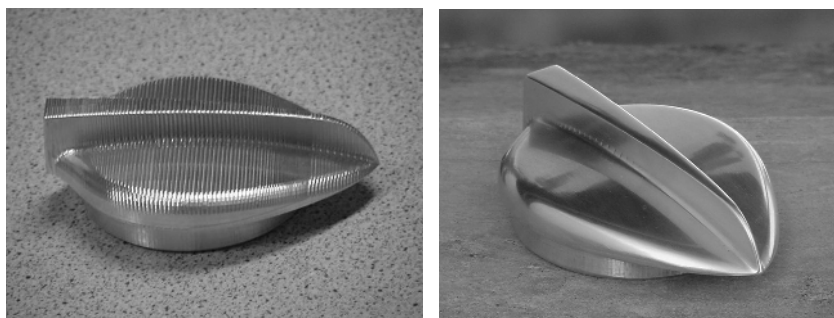
Before the machining itself, the CAD model permits us to design and make one or several machining fixtures, which are necessary to position the part during the machining on the CN machine. After the rebuilding task, CAD models are exported into CAM software. In our case, we use CATIA Machinist from Dassault Système and Visicam

from Missler. The process planning information, tool path of the machining, and the finishing of the part, are then generated taking into account manufacturing parameters (cutting speed, lubrication, etc.) and avoiding any collisions with the environment, see Fig. 9.20.

Once the programs are generated, machining itself is not difficult and it is possible to make a first functional prototype. However, a polishing phase finishes this process, with the objective of creating a physical copy of an original object. As all the previous tasks are computer integrated, this last operation permits the fabrication of both aesthetic and functional characteristics of this radiator cap, see Fig. 9.21.



**Fig. 9.20 Machining of aesthetic surfaces**



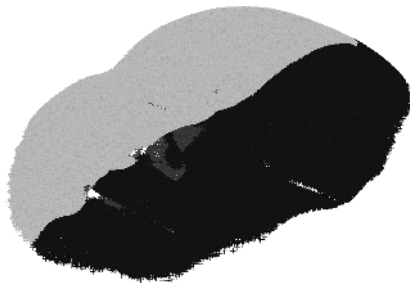
**Fig. 9.21 Before and after polishing**

### 9.3.2 Surface reconstruction using polyhedron meshing

Presented here is an edge-based growing approach to generate a triangular polyhedron model from the point cloud of an object's surface. An original cloud data set is first reduced utilizing the error-based data thinning method. The triangulation process starts with a seed triangle, formed by randomly selecting three neighbouring points. Based on a set of heuristics, the triangular mesh extends outward by continuously jointing suitable external points to it along the boundary edges of the meshed area. A complex free-form surface with holes can be triangulated in one computing session without manually dividing it into several simple patches. A facemask is used for the case study.

#### 9.3.2.1 Part digitization

A non-contact laser-based range sensor was used to collect the cloud data from the facemask. The range sensor (Hymarc, Hyscan 45C) operates on the optical triangulation principle and utilizes a unique synchronized scanning mechanism that provides an effectively wide baseline separation (between source and detector) but encloses all hardware components in a physically compact head. The accuracy specification of the cloud data is  $\pm 0.025$  mm, over a 100-mm depth of field, and a standoff distance of 100 mm. For some complex objects, it cannot be solved in one direction of view. To adequately scan these objects, multiple views must be used. Usually, a typical digitization process requires the range sensor to be repositioned 6–8 times, at various positions and viewing angles, to completely sample the surface patches. In this way, the final cloud data are the result of merging the captured points from different views. For the facemask, the resulting cloud data set, shown in Fig. 9.22, contains approximately 104 000 points.



**Fig. 9.22** The point cloud of the facemask

### 9.3.2.2 Error-based data thinning

Normally, a cloud data set has a very high density of points. A triangulation-based modelling scheme does not necessarily require such a high density of surface points in order to construct an accurate mesh. On the other hand, extremely large-sized cloud data sets could produce long computation times, exceed memory storage limits, and generate computational stability problems. However, arbitrary removal of cloud data points, based on simple data thinning methods, may not ensure that the necessary accuracy between the triangulation and original data is achieved. In the developed algorithm, a spatial filtering method, i.e. the voxel bin method, is first used to reduce the original cloud data file size. The cut-off of the spatial filter is adjustable and permits enough cloud data to be retained so that the reconstruction meets the error criterion. The bin size is first calculated based on a required error tolerance between the original cloud data and the final triangulation. The cloud data are subdivided by the bin size into small cubes that are termed voxel bins. Each bin retains only one point inside. This point is the centre or the nearest point to the centre of the bin. All other points in the bin are removed.

Based on a given error tolerance,  $\varepsilon$ , between the original cloud data and the final triangulation, the bin size is calculated through the following steps:

- Sample  $N$  points from the cloud data file and interpolate a local parabolic quadric surface patch to the  $N$  points by means of least squares fitting.
- Calculate the maximum second partial derivatives from the quadric function coefficients. The maximum triangle edge length ( $\Omega$ ) for a desired error tolerance  $\varepsilon$  can then be calculated by

$$\Omega = 2\sqrt{\frac{2\varepsilon}{(M_1 + 2M_2 + M_3)}} \quad (9.1)$$

where  $M_1$ ,  $M_2$ , and  $M_3$  are the maximum partial derivative values of the locally interpolated surface.

- Every bin has 26 adjoining bins surrounding it; in the 26 bins, 8 bins have one point contacting with the central bin, 12 bins contact along one edge, and the remaining 6 bins contact across a full bin face with it. A triangle edge could pass through two neighbouring



bins that share a common face. A triangle edge could also, in theory, pass obliquely through two neighbouring bins (i.e. at  $45^\circ$ ) that share a common edge. The third situation is the extreme one that a triangle edge could pass through two bins sharing only one point. In this situation, the maximum triangle edge length ( $\Omega$ ) has the following relationship with the bin size ( $B$ )

$$\Omega = \sqrt{3}B \quad (9.2)$$

Combining equations (9.1) and (9.2), we have

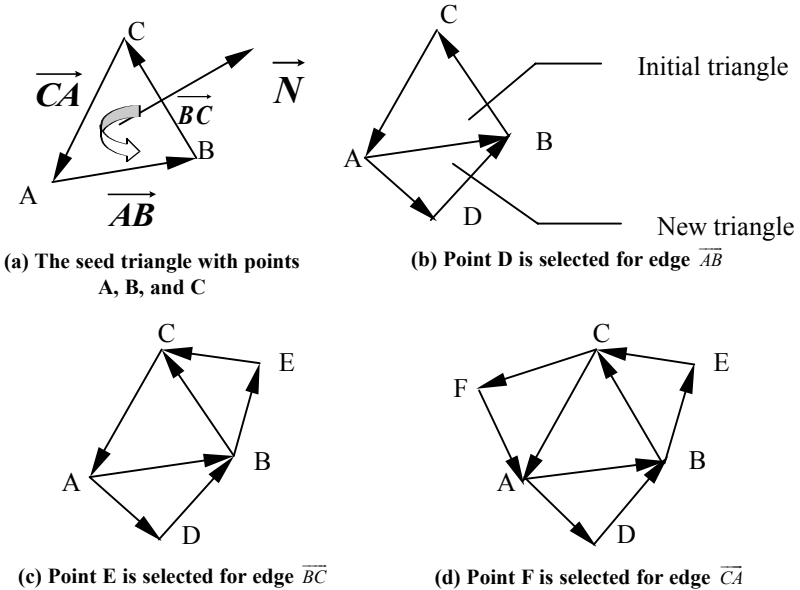
$$B \leq 2 \sqrt{\frac{2\varepsilon}{3(M_1 + 2M_2 + M_3)}} \quad (9.3)$$

### 9.3.2.3 Generation of the triangular mesh

The objective of mesh generation is to cover the reduced point set with a continuous surface mesh of triangular facets. The triangulation algorithm is developed based on a set of heuristic rules. It starts with a selected seed triangle, formed by randomly selecting three neighbouring points. The three edges form the initial boundary edge list of the mesh. The mesh grows by interrogating the geometric and topological information in the neighbourhood of the boundary edge list through application of a set of heuristic rules in order to connect a new point to form a new triangle of the expanding mesh. A neighbouring point to the boundary edge list of the meshed area is selected based on a set of constraints described below. The process is repeated until all the points in the set are meshed. The details of this algorithm are described in the following sections.

#### *The initial triangular facet*

Firstly, three neighbouring points are randomly selected to form the first triangle facet. There should be no point 'inside' this triangle. As shown in Fig. 9.23a, the edges of the triangle are defined as vectors  $\{\overline{AB}, \overline{BC}, \text{ and } \overline{CA}\}$ , following the counter-clockwise direction. The normal of the triangle  $\overline{N}$  points away from the surface patch following the right-hand rule. Every boundary edge forms a vector with a starting and ending point. For instance, edge  $\overline{AB}$  has starting point  $A$  and the ending point  $B$ .



**Fig. 9.23** An example showing the first few steps of triangulation

### *Triangular mesh expansion*

To expand the mesh, a point is selected from the remaining points in the set for each boundary edge. Note that the rules for selecting such a point are described in the sections below. A new triangle is thus formed by a boundary edge and its selected point. A new boundary edge list is then formed for the meshed area, also following the counter-clockwise direction. Take the seed triangle in Fig. 9.23a as an example; point  $D$  is selected for vector  $\overline{AB}$  and a new triangle  $\triangle ADB$  is then formed, shown in Fig. 9.23b. It follows the direction of  $A \rightarrow D \rightarrow B$  and  $\{\overline{AD} \ \overline{DB} \ \overline{BC} \ \overline{CA}\}$ , i.e. the counter-clockwise direction. Similarly, points  $E$  (Fig. 9.23c) and  $F$  (Fig. 9.23d) are selected for  $\overline{BC}$  and  $\overline{CA}$ , respectively. The boundary edge list  $\{\overline{AD} \ \overline{DB} \ \overline{BE} \ \overline{EC} \ \overline{CF}\}$  still follows the counter-clockwise direction. This boundary edge based search and triangle construction process continues until all the points are meshed.

### *Triangulation constraints*

During the triangulation process described above, a new point needs to be identified from the remaining point set in each round. The rules for identifying such a point for a given edge are described as follows.

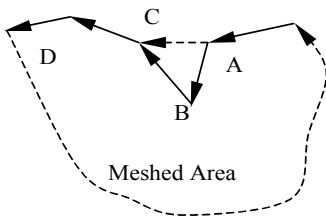
1. *Search Space.* The searching scope for the new point for a boundary edge is limited within the 26 neighbouring voxels of both ending points of the boundary edge.
2. *Minimum triangular facet angle.* The smallest angle within any triangular facet must be larger than the minimum angle parameter,  $\phi$ . This prevents nearly co-linear vertices from forming long, narrow triangular facets. Such facets are unwanted because their surface normal may not accurately match the object's local surface curvature. It was found that  $10^\circ \leq \phi \leq 15^\circ$  generally produced good results.
3. *Maximum angle between two neighbouring facets.* The angle between any two neighbouring triangular facets must be within a specific scope. This angle,  $\theta$ , is defined as the angle between the normal vectors of two neighbouring triangle facets. It must be greater than  $0^\circ$  and less than a specific value. Based on experience, when the specific value is set at  $90^\circ$ , for most free-form surfaces without sharp corners, a good triangulated mesh is achieved. This heuristic helps keep the facet mesh growing smoothly.
4. *Being nearest to the middle of the edge.* There may be more than one point satisfying all the rules mentioned above, but only one point should be selected. Based on experience, the one which is the nearest to the middle of the boundary edge at hand, can lead to achieving an evenly distributed triangular mesh.

In every step of forming a triangle, only one point satisfying the constraints described above is selected to form a new triangle. This point is called the 'suitable point' and the edge used in the search is called the 'searching edge'. The location of a suitable point may fall in the following situations.

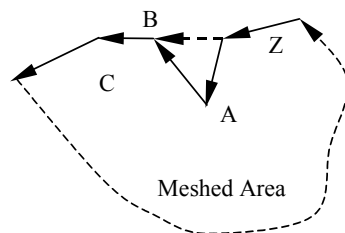
1. There is no suitable point. This means the triangulation reaches the boundary of the image (internal or external).
2. The suitable point is within the boundary of the image. This is the most common situation in triangulation.
3. The suitable point is on the boundary and next to the ending or starting point of the searching edge in the current external edge loop ( $C$  to  $AB$  in Fig. 9.24a and  $Z$  to  $AB$  in Fig. 9.24b). In Fig. 9.24a, the original counter-clockwise external edge loop is  $\dots \rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow \dots$ , meaning that there is no link between  $A$  and  $C$ . In this case,  $C$  is found to be the suitable point for searching edge  $\overline{AB}$ , and a new triangle  $\triangle ACB$ . The resulting new external edge

loop becomes  $\dots \rightarrow A \rightarrow C \rightarrow D \rightarrow \dots$ , thus filling the gap between  $A$  and  $C$  in the original boundary list. Similarly, the gap between  $B$  and  $Z$  in the original boundary list is filled up in Fig. 9.24b.

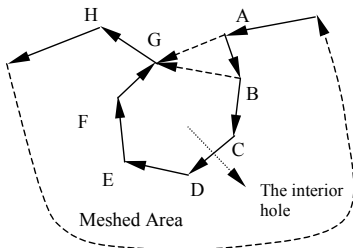
4. The suitable point is on the boundary, but not next to the searching edge in the current external edge loop ( $G$  to  $AB$  in Fig. 9.24c and  $W$  to  $AB$  in Fig. 9.24d). In Fig. 9.24c, the current external edge loop is  $\dots \rightarrow A \rightarrow B \rightarrow \dots \rightarrow F \rightarrow G \rightarrow H \rightarrow \dots$ ,  $G$  is found to be the suitable point to searching edge  $\overline{AB}$ , and a new triangle  $\Delta AGB$  is formed. Then  $\dots \rightarrow A \rightarrow G \rightarrow H$  becomes the counter-clockwise external boundary loop. On the other hand,  $B \rightarrow C \rightarrow D \rightarrow \dots \rightarrow E \rightarrow F \rightarrow G \rightarrow B$  falls into a clockwise direction and is no longer changed further. This loop is separated from the external boundary loop and is called an internal loop. Actually, this is the hole inside this surface patch. For the case shown in Fig. 9.24d, the original external edge loop is  $\dots \rightarrow V \rightarrow W \rightarrow X \rightarrow \dots \rightarrow A \rightarrow B \rightarrow C \rightarrow \dots$  and  $W$  is found to be the suitable point of searching edge  $\overline{AB}$ , and a new triangle  $\Delta AWB$  is formed. It can be seen that the vector  $\overline{AW}$  forms a closed clockwise loop of  $X \rightarrow Y \rightarrow Z \rightarrow A \rightarrow W \rightarrow X$ . This internal loop is thus separated



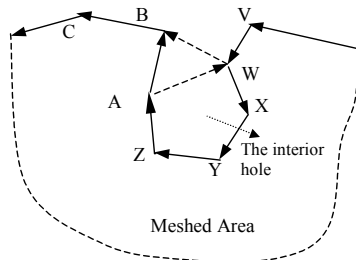
(a) The new point  $C$  is on the boundary and next to the searching edge,  $\overline{AB}$ , in the external edge loop



(b) The new point  $Z$  is on the boundary and next to the searching edge,  $\overline{AB}$ , in the external edge loop



(c) The new point  $G$  is on the boundary, but not next to the searching edge,  $\overline{AB}$ , in the current external edge loop



(d) The new point  $W$  is on the boundary, but not next to the searching edge,  $\overline{AB}$ , in the current external edge loop

**Fig. 9.24** Triangulation when the suitable point is on the boundary

from the external loop resulting in a hole. The new external boundary loop becomes  $\dots \rightarrow V \rightarrow W \rightarrow B \rightarrow C \rightarrow \dots$

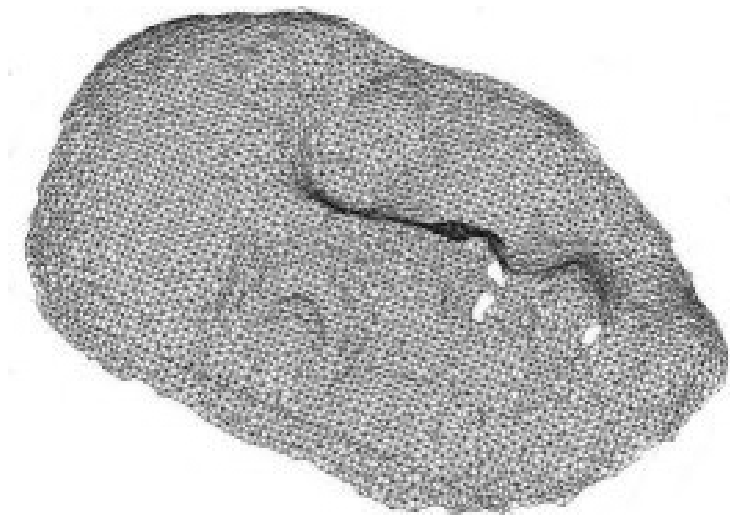
### *The polyhedron model of the facemask*

This error-based triangulating method has been implemented in the Unigraphics environment employing its internal programming tools. To test the efficacy of the developed technique, two case studies are presented here. The first case study is of the mask, which is composed of four range data patches referenced to a single datum point on the bed of the CMM. The facemask is an intricate free-form surface occupying a volume of 250 mm  $\times$  200 mm  $\times$  60 mm and the surface possesses smooth regions interrupted by sharp features with high surface curvature. The surface is problematic for B-spline surface fitting due to the difficulties inherent in knot vector creation, data point parametrization, and control point location. The binning and triangulation algorithm was applied to the original data file containing 104 216 points. Employing an error tolerance of 0.4 mm a commensurate bin dimension of 3.0 mm was determined. Applying this spatial binning criterion to the data, and meshing the result, the polyhedral model shown in Fig. 9.25 was obtained. This triangulation contains 5410 vertices. Three holes on the original point set have been meshed with a smooth boundary.

### *Fabrication using SLA and CNC machining*

Milling, using CNC methods, can be performed from the surface definition generated by this technique. For example, computer-aided manufacturing packages, such as the DelCAM Duct program, can utilize the polyhedral surface representation. Free-form surface tool path generation methods, employing non-isoparametric means, are ideally suited to the polyhedral triangular facet model. In particular, researchers investigating tool-part collision and gouge-free machining of free-form surfaces employ this description.

For the facemask, the polyhedron model was used for fabrication through two manufacturing techniques: RP and five-axis machining. For the former, the model is easily converted to an STL format that can be used by an SLA machine directly. The generated model is shown in Fig. 9.26. For the latter, an in-house developed program was used to generate the tool-path (collision and gouging free) and the part is machined on a five-axis machine (Deckel Maho DMU 50V), see Fig. 9.27.



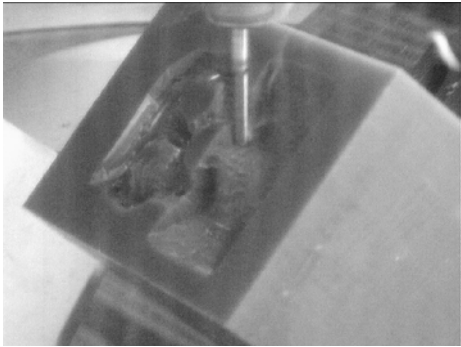
**Fig. 9.25 The triangular mesh model of the facemask**



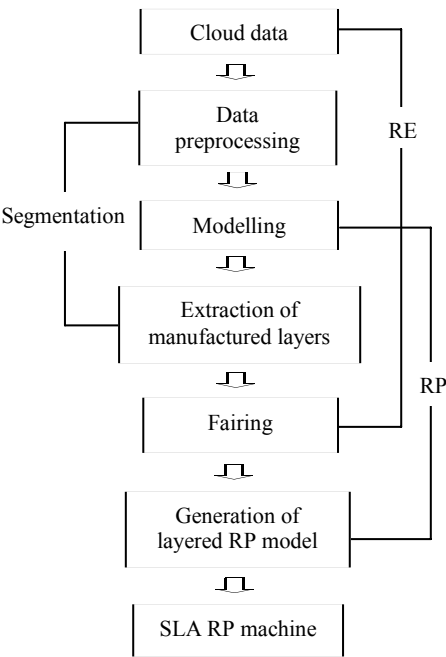
**Fig. 9.26 The facemask produced by SLA**

### **9.3.3 Rapid fabrication using a point-based segmentation approach**

This section introduces a novel point-based curve representation, from which the RP manufactured layers can be directly extracted. An error-based segmentation algorithm is also developed to accomplish direct RP manufacturing of cloud data. In this process, neither a surface



**Fig. 9.27 Five-axis machining of the facemask**



**Fig. 9.28 The point-based segmentation method**

model nor an STL file is generated. The overall procedure is illustrated in Fig. 9.28. This is accomplished with three main stages, i.e. data pre-processing (data subdivision and reduction), data modelling (IPCM construction), and extraction of the manufactured layers.

### 9.3.3.1 Data pre-processing

Here, a new highly efficient method is developed for cloud data reduction, in which cloud data are adaptively subdivided into many regions and then are reduced by individually compressing the data in each region.

As cloud data are a dense data set, if the subdivision accuracy is well controlled, the shape represented by the points in each region can be considered as one three-dimensional curve (continuous) or multiple three-dimensional curves (discontinuous). For a three-dimensional curve, its basic shape is exhibited by the feature points (FPs): corners or high curvature points. Hence, the objective of data reduction is to search and find all the FPs in the region and remove the rest. Randrup (41) has employed a digital image reduction technique to process projection data for ruled surface approximation, in which the skeleton of the digital image can be obtained to construct the necessary directrix curves. This method is adopted to perform the data reduction for each region.

#### *Data subdivision*

The aim of data subdivision is to convert the three-dimensional cloud data into a set of two-dimensional regions by adaptively subdividing the cloud data with a cluster of parallel planes. This cluster of subdivision planes is defined according to a user-specified reference plane. When the subdivision planes are defined, the cloud data are first subdivided into several initial regions based on the defined reference plane  $T$  and its normal direction  $\hat{\mathbf{n}}$ . The user determines these after examining the visually observable shape of the cloud data. Then, for each region of the data set, a new subdivision plane passing the central position of the region is generated. This plane is also treated as the *projection plane* for the data points of the region. Denoting the region of the data set and the corresponding central position as  $\mathbf{P}$  and  $\mathbf{O}$ , respectively, the distance from a given point in the region,  $\mathbf{p} \in \mathbf{P}$ , to the projection plane  $T$ , can be expressed as:

$$\text{dist}(\mathbf{p}, T) = \|\hat{\mathbf{n}}^T \cdot (\mathbf{p} - \mathbf{O})\| \quad (9.4)$$

We estimate the average projection error of the data points in the region to the projection plane with the following error indicator:



$$ave\_proj\_err = \left| \frac{1}{n} \sum_{p \in P} (dist(p, T))^2 \right|^{1/2} \quad (9.5)$$

where  $n$  is the total number of the points in the region. In the subdivision algorithm, if the calculated average projection error of the points in the region is greater than a given subdivision error, this region is further subdivided into two halves from the central position of the region. This subdivision process continues for each subdivided region until all the average projection errors of the regions are within the desired subdivision error. Thus, the original three-dimensional cloud data are converted to a set of two-dimensional subdivided regions.

### *Data reduction*

For the data points of each subdivided region, we consider a grid structure in the projection plane. Each square in this grid is associated with the value 1 (*black point*) if the intersection between the square and the projection of the point in the region is not empty. Otherwise it is 0 (*white point*). In practice, it is difficult to determine a desirable square size of the grid (1). If the square is too small, the digital plane can have holes (white points among black points), which may lead to a complex reduction result. Hence, to ensure that the digital plane has no holes, the size of the square is normally chosen to be sufficiently large. In this way, the digital plane of the projection plane mapping to the three-dimensional points in the region is constructed.

Data reduction is firstly employed to derive a skeletal curve from the black points in the digital plane by removing all the black points that are *deletable*. Here, we employ 28 3×3 thinning templates proposed by Jang and Chin (42) to compress the data of each region, which is a parallel thinning method. The 28 thinning templates define 28 kinds of black points that can be removed from the digital plane. In the thinning process, black points in the digital plane are firstly compared with the templates. Once a compared black point matches one of the defined templates, the black point is removed (to be a white point). The comparison is then resumed. This process iterates until no deletable black points can be identified from the digital plane with the defined thinning templates. Let  $S$  and  $\Gamma$  denote a digital plane and a thinning template, respectively. Assuming that the deletable black point defined by  $\Gamma$  is  $P$ , and a black point in  $S$  is  $p$ , the proposed thinning operation can be expressed as follows:

$$\bigcap_{p \in P} S + \{-p\} \subseteq S \quad \text{with} \quad P \in \Gamma \quad \text{and} \quad p \in S \quad (9.6)$$

In practice, to improve the efficiency and robustness of the thinning algorithm, the defined 28 thinning templates are classified into three levels, i.e. the digital plane is compressed using the templates with a priority of  $\Gamma_1 > \Gamma_2 > \Gamma_3$ . Compared with the reduction method used by Randrup (41), our proposed method is more efficient in thinning the data and is more robust.

When all the deletable black points in the digital plane have been removed, a skeleton of the digital image is obtained. We select three-dimensional points in the region that correspond to the centres of the black points (black squares) and define them as the FPs of the region. Based on the two-dimensional structure of the black squares of the skeleton in the digital plane, the selected FPs are ordered and connected with straight-line segments. A three-dimensional *skeletal curve* is therefore generated. The generated skeletal curve can be used to roughly cover the basic shape of the region because some of the important shape points of the region may not be included in these selected FPs. This is due to the determination of the square size of the grid for the digital plane. Hence, a maximum-distance-error (MDE) method is proposed to restore the potentially important shape points from the rest of the points in the region. These new retrieved shape points by MDE, together with the previously selected FPs, are all considered as the FPs of the region. The shape of the region is then depicted with these FPs.

### 9.3.3.2 IPCM construction

For modelling, we propose a novel point-based curve model (IPCM) instead of a more commonly used triangular mesh model, which is simpler but effective and more computationally efficient. As the reduced cloud data are a set of regions of FPs, the IPCM is constructed across the direction of the regions. The data structure for the IPCM is defined as follows:

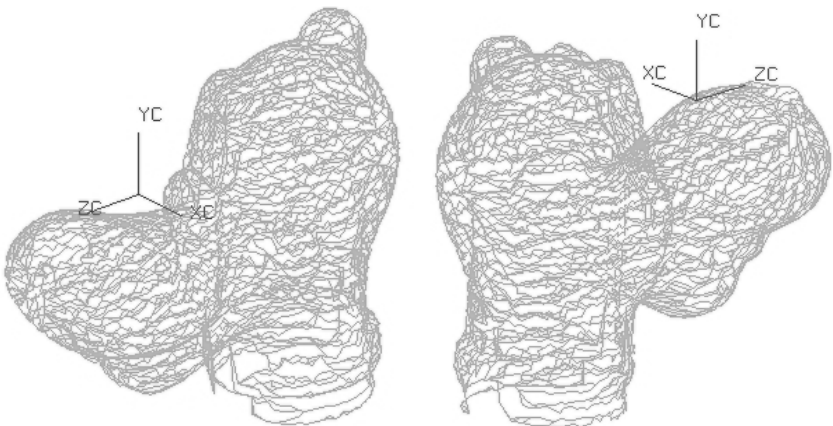
```
typedef struct ipcm_pt
{
    int          point_id, region_id;
    int          status;
    double       *point3D;
    ipcm_pt     *head, *tail;
    ipcm_pt     *front, *next;
}*IPCM_PT;
```

In this data structure, pointers *head* and *tail* are used to record the FPs in the same region, while pointers *front* and *next* are used to connect FPs in different regions to generate the IPCM. In the modelling process, the shortest distance of two points is applied to determine the two neighbouring FPs in different regions. The construction of the IPCM starts from one of the end regions (the front most or the back most) and is a progressively growing process. This growing process consists of two stages: the first is across the direction of the regions, and the second (reversed growing) is in the reversed direction across the regions. These two growing processes are carried out region by region. The first growing process checks all the FPs and searches for the neighbouring FPs by retrieving all the other FPs. The reversed growing process only checks FPs whose neighbouring FPs are still empty after the forward growing.

Compared with the conventional triangular mesh model for cloud data, the proposed IPCM model is much simpler but effective when it is applied to extract layer contours for RP. An example of IPCM that is constructed from carton toy cloud data is illustrated in Fig. 9.29. In this example, the cloud data are subdivided with a cluster of plane  $XcYc$  and the IPCM is constructed along the direction of  $\pm Zc$ .

### 9.3.3.3 Extraction of manufactured layers

The RP layer contours are extracted by intersecting the constructed IPCM with a set of user-defined planes (slicing planes) that are parallel and uniformly distributed. The number of RP layers is determined



**Fig. 9.29** An example of IPCM and its two different views

according to a user-given layer thickness, which corresponds to the allowable building thickness in RP. Suppose that the cluster of slicing planes (intersection planes) is positioned with a sequence of  $z$ -co-ordinates:  $\mathbf{Z} = (z_0, z_1, \dots, z_n)$  and  $z_0 < z_1 < \dots < z_n$ , the value of  $z_k \in \mathbf{Z}$  is determined based on the maximum and minimum  $z$ -co-ordinates of the FPs in the IPCM. Denoting the user-given layer thickness and the maximum and minimum  $z$ -co-ordinates of the IPCM as  $z_{\text{pitch}}$ ,  $z_{\text{min}}$ , and  $z_{\text{max}}$ , respectively, the value of  $z_k \in \mathbf{Z}$  is calculated as given below:

$$z_0 = z_{\text{min}}, \quad z_n = z_{\text{max}}, \quad z_k = z_0 + i \cdot \left\lfloor \frac{z_n - z_0}{z_{\text{pitch}}} \right\rfloor \quad \text{with } i = 1, \dots, n-1 \quad (9.7)$$

The intersection between the slicing planes and the constructed IPCM is determined by comparing the sequence  $\mathbf{Z}$  with the  $z$ -co-ordinates of the FPs and their neighbouring FPs in the IPCM. The intersection points (or the layer contour points) are calculated by linearly interpolating the FPs and the corresponding neighbouring FPs. For example, supposed that the  $z$ -co-ordinate of an FP and its neighbouring (front neighbouring or next neighbouring) FP is  $z_{\text{FP}}$  and  $z_{\text{neighbouring}}$ , respectively, if there exists a value  $z_k \in \mathbf{Z}$ , such that  $z_{\text{neighbouring}} \leq z_k \leq z_{\text{FP}}$  or  $z_{\text{FP}} \leq z_k \leq z_{\text{neighbouring}}$  is satisfied, there exists an intersection between the slicing plane and the IPCM, and this intersection point is calculated by linearly interpolating the mentioned two FPs.

In the algorithm, we can determine the intersections by respectively grouping the FP and its front neighbouring FP and the FP and its next neighbouring FP, and then comparing these two groups with the  $\mathbf{Z}$  sequence. However, this may produce some repeated intersection points. Hence, a *status* value of either 1 or 0 is applied in IPCM data structure to avoid the above-mentioned repetition. The proposed extraction algorithm is described as follows:

1. *Initialization.* Retrieve all the FPs of the IPCM by checking whether the FP and its neighbouring FP' satisfy the following condition:

$$\begin{cases} \text{FP} \rightarrow \text{next} = \text{FP}' \\ \text{FP}' \rightarrow \text{front} = \text{FP} \end{cases} \quad \text{or} \quad \text{FP} \rightarrow \text{next} = \text{NULL}$$

If the above condition is satisfied, the *status* value of the FP is initialized as  $\text{FP} \rightarrow \text{status} = 0$ , otherwise  $\text{FP} \rightarrow \text{status} = 1$ . For

FP→status = 0, only the FP and its front neighbouring FP are used to compare with the  $Z$  sequence of the slicing planes to determine whether intersections exist between them, while for FP→status = 1, both the FP and its front neighbour, and the FP and its next neighbour, are respectively compared.

2. *Calculate the layer contour points (intersection points).* For each slicing plane, all the FPs of the IPCM are retrieved and checked with the aforementioned method to determine whether an intersection exists. Hence, the computation complexity of the proposed algorithm is  $O(n - m)$ , where  $n$  is total number of slicing planes and  $m$  is the total number of the FPs in the IPCM. Suppose that an intersection is determined between plane  $z_k \in Z$  and the IPCM, the identified two FPs are  $FP_i^{(R_m)}$  and  $FP_i^{(R_n)}$ , and the corresponding  $z$ -co-ordinates are  $z_m$  and  $z_n$ ; the intersection point  $P_i^{(Z_k)}$  for the RP layer  $z_k$  is then calculated as follows:

$$P_i^{(Z_k)} = \frac{Z_k - Z_m}{Z_n - Z_m} FP_i^{(R_n)} + \frac{Z_n - Z_k}{Z_n - Z_m} FP_i^{(R_m)} \quad (9.8)$$

3. *Generate the layer contours.* The intersection points having the same  $z$ -co-ordinates correspond to the contour points of an RP layer. These two-dimensional points can be sorted and connected with straight-line segments by calculating the shortest distance between the points. An RP layer is then generated. However, due to the processing errors in the segmentation, the extracted contour/contours of each RP layer may not be sufficiently smooth. Hence, a discrete curvature-based fairing algorithm has been developed to automatically smooth these layer contour/contours (43).

The intersection planes are user-defined based on the constructed IPCM. To achieve a satisfactory RP part, the intersection direction should be defined to achieve sufficient intersection between the intersection planes and the IPCM. As the IPCM is constructed across the direction of the regions, we should avoid this direction to be defined as the intersection direction because it leads to a minimum of intersection points between the IPCM and the intersection planes.

#### 9.3.3.4 Direct RP manufacturing

Direct RP manufacturing involves building the part directly from the cloud data without constructing a surface representation and generating

an STL model. Here, this is implemented with the extracted manufactured layers. However, the faired manufactured layers cannot be directly sent to RP machines for fabrication because the extraction cannot ensure all the layer curves are closed. Hence, each manufactured layer curve should be checked and closed before the fabrication. We termed the manufactured layer curves that are faired and closed as the layer-based RP model. This model is served as a direct CAD input to the RP machine.

The algorithm for the direct RP manufacturing of cloud data has been implemented with C/C++ on an HP-C200 workstation in the Unigraphics environment. Here, a case study is presented to illustrate the efficacy of the proposed algorithm. The experimental RP machine is chosen to be Sony Solid Creator JSC2000 SLA machine, which is controlled with a NWS-3865 network station and the laser power is Coherent Innova 300. In the experiment, the intensity of the laser power is set as 0.250 W and the prototyping material is SCR-310 photo polymer. The generated layer-based RP model is directly fed to the SLA machine for prototype manufacturing.

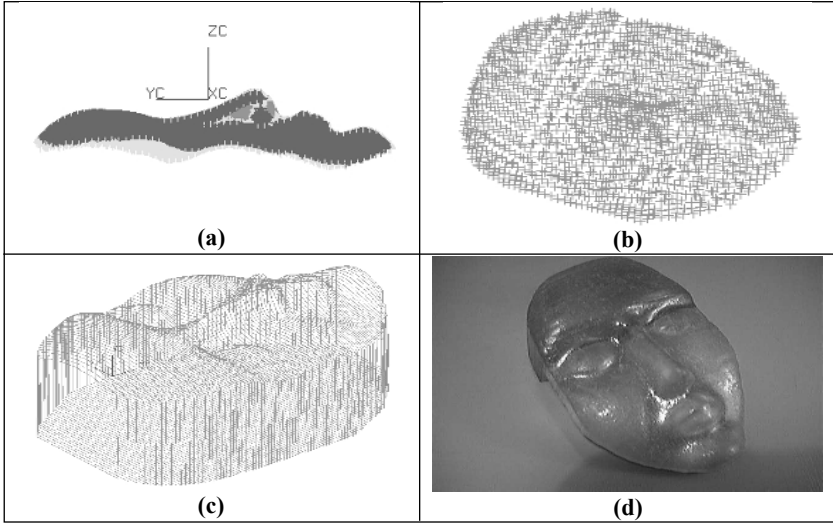
The part used in the case study is the facial mask, see Fig. 9.30. The parameter to define the subdivision planes and the intersection planes was chosen as (plane  $ZcYc, \pm Xc$ ), as shown in Fig. 9.29a. Based on this parameter, the cloud data are compressed to construct an ICM and a closed layer-based RP model is generated, as shown in Figs 9.30b and c. The shape accuracy and the layer thickness for the final testing RP model are chosen to be 0.25 mm and 0.3 mm, respectively. The total number of extracted RP layers is 807 and the fairing of these layers took about 66 min. The faired and closed RP model is directly fed to the SLA machine and the prototyping time is approximately 18 h. Figure 9.30d shows the final RP product.

## 9.4 VERIFICATION

In the following sections, we propose a solution to automatically verify and inspect the manufacturing part by comparing the digitized point cloud and the CAD model.

### 9.4.1 Introduction

In recent years, many technological improvements have been made in all the different parts of conception and machining of a product.



**Fig. 9.30 Reverse engineering of the facemask using the segmentation method**

Three-dimensional digitizing of complex free forms is one of these evolutions. It has become a very important part of the reverse engineering process. However, the installation of a digitizing cell in an industrial environment is not straightforward and requires the solving of many technical problems.

The integration of a laser sensor on the CMM of the Atelier Inter-établissement de Productique Lorrain allowed us to highlight these problems. They were the starting points of several works. These works are part of a global project managed in the Research Centre for Automatic Control of Nancy (CRAN, France). Stéphane Davillerd began with the integration of the KLS 51 sensor from KREON Industrie (France) on a CMM from DEA. He also developed a simulation and off-line programming module in the SILMA Cimstation Robotics environment. Then he proposed a state-of-the-art about the different three-dimensional digitizing technologies. Benoît Sidot worked on the determination of the process planning of digitalization: which are the different positions needed to completely digitize the mechanical part? To answer this question, he worked on a visibility calculation algorithm, including our sensor field of view.



### 9.4.2 State of the art

In his PhD work, Prieto (43) draws up an assessment of works on automatic digitizing, using a plane laser sensor. His method (44) gives some good results and the originality is the definition of a confidence coefficient for the result of the scanning. This coefficient depends on the difference between the optimal position of the sensor and the possible ones. Actually, the more the laser beam is normal to the surface, the better is the quality of the point clouds. This approach was tested for parts made up of planes and cylinders. This should be extended for more complex parts.

### 9.4.3 The CRAN project

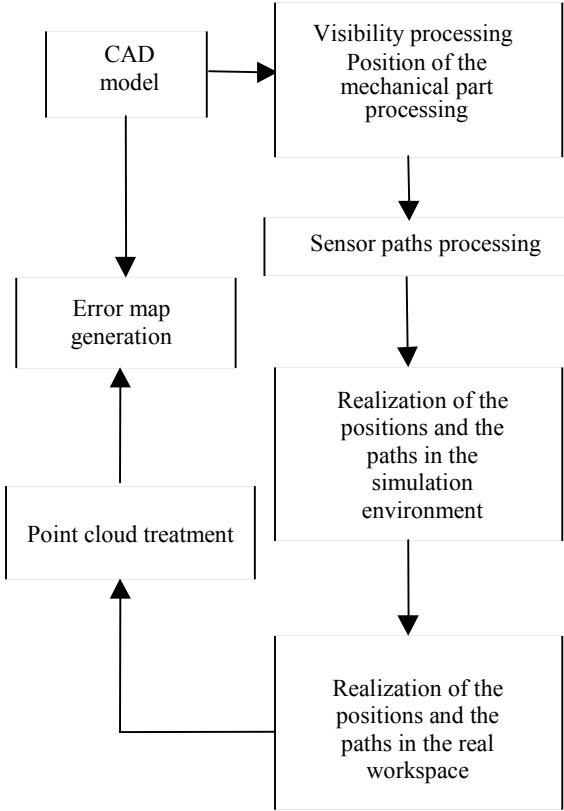
This project comes from a study concerning the integration of a three-dimensional digitizing tool in a context of industrial production. The main idea is to propose a system to a company. This system permits comparison of the CAD model of a mechanical part and the point cloud from the scanning of this part. Our research work concerns the dimensional control of the mechanical part and not, for the moment, the reverse engineering, see Fig. 9.31. This project was born with the acquisition of a KLS 51 sensor from KREON Industrie. We also have a gantry-type CMM from DEA. A rotary table on the marble is used as a fourth axis.

Finally, we want to have an error map by comparing the CAD model of the mechanical part with the point cloud obtained by the automatic scanning of this part. Our work is concerned with the scanning automation. In fact, the point cloud treatment and the error map generation are already done by software like Surfacr by Imageware or Polyworks by Innovmetric. The simulation environment has been developed on Silma CimStation software. It allows us to test the trajectories and even to do off-line programming.

### 9.4.4 Digitization process planning

This is the second step of the project. We had to define a method and an algorithm for the determination of the scanning strategy, which consists in the sensor and the mechanical part positioning in a known environment. The retained method to solve the three-dimensional digitizing strategy generation is based on the concept of visibility. This one has already been used in some areas like milling, control, foundry, vision, and infography.





**Fig. 9.31 Methodology**

The phase of research of the visibilities is based on spherical geometry and on a method specially developed for the technology of our sensor (46).

#### 9.4.4.1 Visibility calculation

To calculate the visibilities, we use a CAD model of the shape we want to scan which is discretized. Logically, we choose the STL format. In this one, the surface of our shape is defined by triangles, see Fig. 9.32.

We define a unitary sphere as a discretized sphere, see Fig. 9.33.

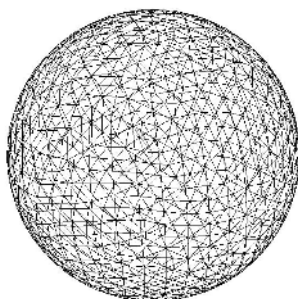
There are three steps in the visibility determination. First, we have to determine the local visibility. Let us consider a triangle  $i \in [1:n]$  from

```

solid rainure_50
  facet normal 0.000000 0.000000 1.000000
    outer loop
      vertex 20.000000 40.000000 70.000000
      vertex 40.000000 40.000000 70.000000
      vertex 40.000000 80.000000 70.000000
    endloop
  endfacet
  facet normal 0.000000 0.000000 1.000000
    outer loop
      vertex 0.000000 0.000000 120.000000
      vertex 20.000000 0.000000 120.000000
      vertex 20.000000 40.000000 120.000000
    endloop
  endfacet
  ....

```

**Fig. 9.32 STL file**



**Fig. 9.33 Discretized sphere**

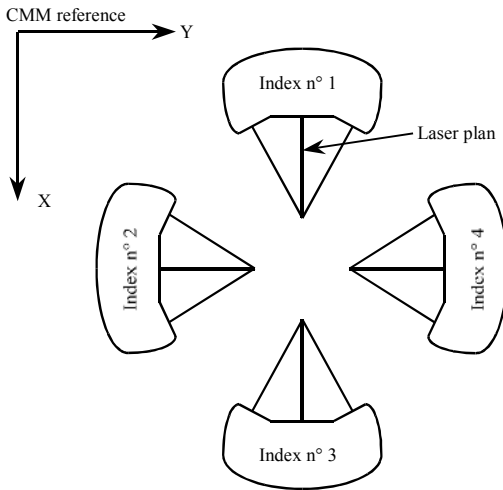
the surface of the mechanical part, and a triangle  $j \in [1:m]$  from the unitary sphere. We say that the triangle  $i$  is visible in the direction  $j$  if the angle between  $n_i$  and  $n_j$  (respectively the normal of the triangle  $i$  and the triangle  $j$ ) is less than  $84^\circ$ . In fact, the local visibility of a triangle  $i$  from the part is a hemisphere limited by the  $84^\circ$  condition. This condition was obtained by experiment.

After this, we have to compute the global visibility. We look for the local visibility directions that intersect another triangle of the part surface. We obtain a restriction of the local visibility map which is the map of all the possible visibility directions of the unitary sphere for our laser sensor.

Finally, we must determine the real visibility. To do that, we take into account some additional constraints like the geometrical shape of the sensor, the visibility by the laser beam, and at least one of the two CCD cameras.

#### 9.4.4.2 Digitizing strategy

In fact, there are now three possibilities. In the first case, we can scan the part for only one position of the sensor and one position of the part. Here, all the triangles of the part are visible from at least one direction of the real visibility map. In the second case, we use one position of the part but several of the four possible positions of the sensor, see Fig. 9.34. In this case, we create an STL file for each used sensor position (STL\_index\_1.stl ...). In the last case, we must use a rotary table to have several positions of the part and several of the four possible positions of the sensor.



**Fig. 9.34 The four possible directions of the sensor**

For all these cases, our algorithm gives us four rotations for the part. The first is  $\beta$  angle rotation around the Z-axis.

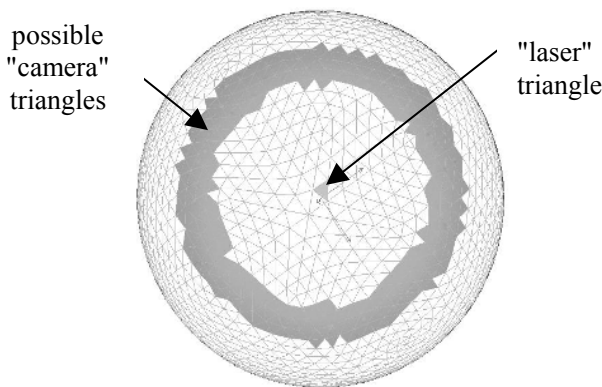
$$\begin{pmatrix} \cos \beta & -\sin \beta & 0 \\ \sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The second is an  $\alpha$  angle rotation around the  $Y$ -axis, and the third is a  $\delta$  angle rotation around the  $Z$ -axis.  $\alpha$ ,  $\beta$ , and  $\delta$  angles come from the visibility calculation. To be scanned, a triangle of the part surface must be visible by the laser beam and at least one of the two cameras. So, a triangle of the part surface must be visible in at least two directions of the unitary sphere. On the discretized sphere, we associate two 'camera' triangles to the 'laser' triangle according to the geometrical shape of the sensor, see Fig. 9.35.

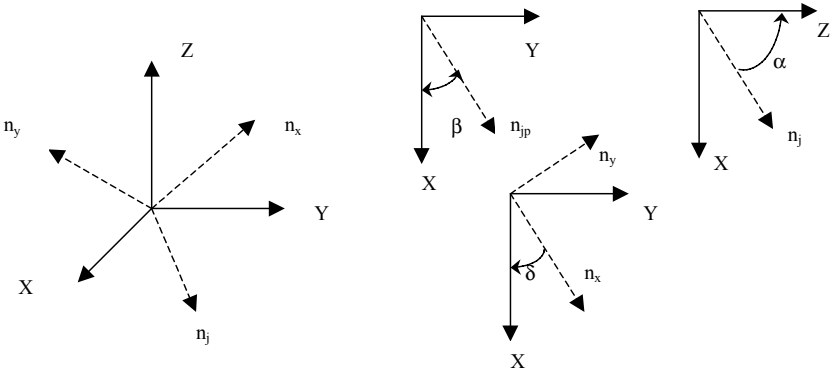
The three rotations are those, which are necessary to put into correspondence a reference frame created from the normal vectors of the three triangles (two for the cameras and one for the laser), and the reference frame related to the centre of the field of measurement of our sensor.

The  $\beta$  angle is between the  $X$ -axis of the sensor's field of measurement reference frame and the  $n_j$  projection on the  $XY$  plane. The  $\alpha$  angle is between  $n_j$  and  $Z$ . In the end, the  $\delta$  angle is between  $n_x$  and  $X$ , see Fig. 9.36.

There is a fourth and last rotation of a  $22.5^\circ$  angle around  $Y$ . This is the angle between the sensor's field of measurement reference frame and the machine reference frame when the sensor is in the index 1 position, see Figs 9.34 and 9.37.

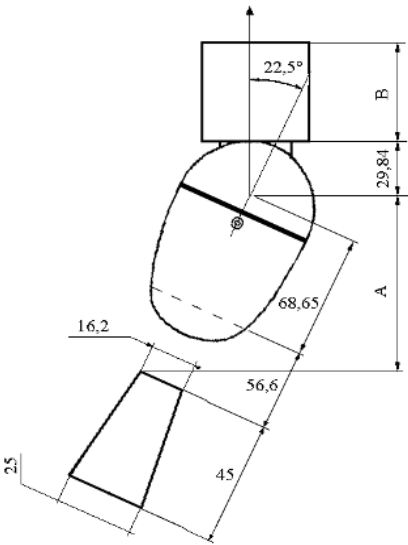


**Fig. 9.35 Laser and camera triangles**



$n_{jp}$  is the  $n_j$  projection on the  $XY$  plane

**Fig. 9.36** The two reference frames



**Fig. 9.37** Sensor and its field of view

**9.4.5 Path planning**

Now, we have the position of the piece and we know which triangles are visible (can be scanned). An STL file was created with these triangles for each index position. For example, all the triangles visible for index 1 position are in the file STL\_index\_1.stl.

In the following sections, we explain our method for scanning in index 1 position. It is the same for the other index positions with little differences concerning the axis of the  $22.5^\circ$  rotations done just before and after the trajectory calculations. These ones are here to put  $Z_f$  from the field of view reference frame, in the same direction of  $Z$ . This operation makes the following calculations simple. From this file (STL\_index\_1.stl), we extract all the point co-ordinates to obtain a matrix:

$$\begin{pmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ X_3 & Y_3 & Z_3 \\ \dots & \dots & \dots \\ X_n & Y_n & Z_n \end{pmatrix}$$

We have to put this point cloud in position. We compute this matrix with the four-rotations matrix to obtain the point co-ordinates in the final piece orientation. After this, we order the points by  $X$  ascending (or  $Y$  if we are in index 2 or 4 positions). Now we can begin to calculate the trajectories. We apply a  $-22.5^\circ$  rotation around the  $Y$ -axis. We are in the following situation, see Fig. 9.38.

The relative position between the field of measurement (field of view) and the point cloud is satisfactory. The aim of the last rotation is to simplify our calculations by working in a field of view reference frame that is the same as the machine reference frame. We cut the point cloud

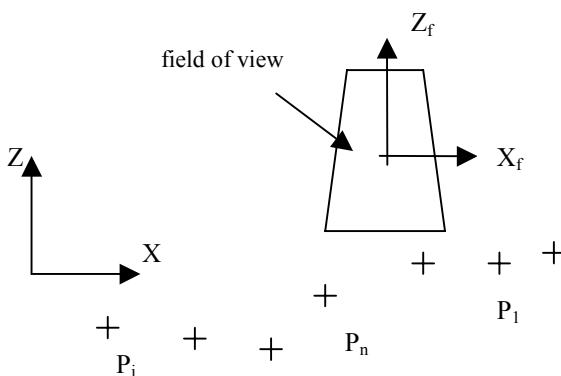


Fig. 9.38 Point cloud and field of view

in a set of stripes. The width of the stripes is related to the width of the field of measurement, see Fig. 9.39. This method has already been used by Chang and Fengfeng (47).

As our points are ordered by  $X$  ascending, we identify  $X_{\max}$  and  $X_{\min}$ . We put into a table all the points, which verify:

$$(1) \quad X_{\max} - 18 < X < X_{\max}$$

We order the resulting table by  $Y$  ascending and we identify  $Z_{\max}$  and  $Z_{\min}$ .

IF

$$Z_{\max} - Z_{\min} < 45$$

THEN

$$\text{traj} = \left( X_{\max} - 9; Y_{\min} - 5; \frac{Z_{\max} + Z_{\min}}{2} \right) \longrightarrow$$

$$\left( X_{\max} - 9; Y_{\max} + 5; \frac{Z_{\max} + Z_{\min}}{2} \right)$$

ELSE

$$\text{traj} = (X_{\max} - 9; Y_{\min} - 5; Z^*) \longrightarrow (X_{\max} - 9; Y_{\text{next}}; Z^*) \longrightarrow \dots$$

$$\dots \longrightarrow (X_{\max} - 9; Y_{\max} + 5; Z^*)$$

with  $Z^*$  the  $Z$  average for a given  $Y$ . In fact, if several points have the same  $Z$ , we replace the piloted point by the  $Z$  average of all these points. The trajectory begins 5 mm before the  $Y_{\min}$  point and finishes 5 mm after the  $Y_{\max}$  point.

Then we make another stripe:

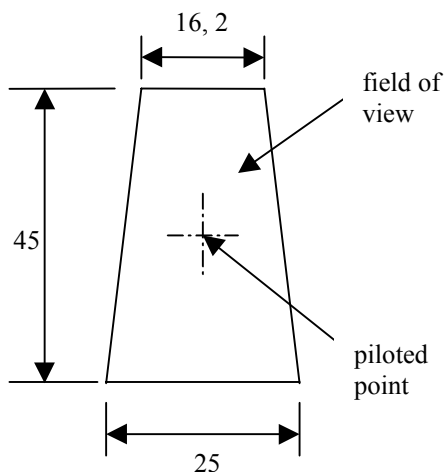
$$X_{\max} = X_{\max} - 15$$

IF

$$X_{\max} \leq X_{\min}$$

THEN stop

ELSE go to (1)



**Fig. 9.39 Field of view**

In fact, we have just cut the point cloud in a set of parallel stripes. These ones are parallel to the  $Y$ -axis and their width is 18 mm. Each stripe oversteps by 3 mm the preceding one. The width of 18 mm is a parameter that the user can modify if the topology of the surface needs it. Once all these operations have been done, we have a set of trajectories:

traj1 = P11 ; P12 ; ... ; P1 $i_1$

traj2 = P21 ; P22 ; ... ; P2 $i_2$

...

traj $k$  = P $k$ 1 ; P $k$ 2 ; ... P $k$  $i_k$

To obtain the entire trajectory, we must link the last point of traj1 with the last point of traj2, the first point of traj2 with the first of traj3, etc. To have the co-ordinates of the trajectory in the machine reference frame, we have to perform a last rotation of  $22.5^\circ$  around  $Y$ .



To have the co-ordinates of the trajectory in the machine reference frame, we have to perform a last rotation of  $22.5^\circ$  around  $Y$ . We have now the trajectories for the scanning of the piece from a single origin of the machine reference frame. So we must know the location of the piece and enter an offset in the three axes of our CMM to identify the final co-ordinates of the trajectory.

9.4.6 Application

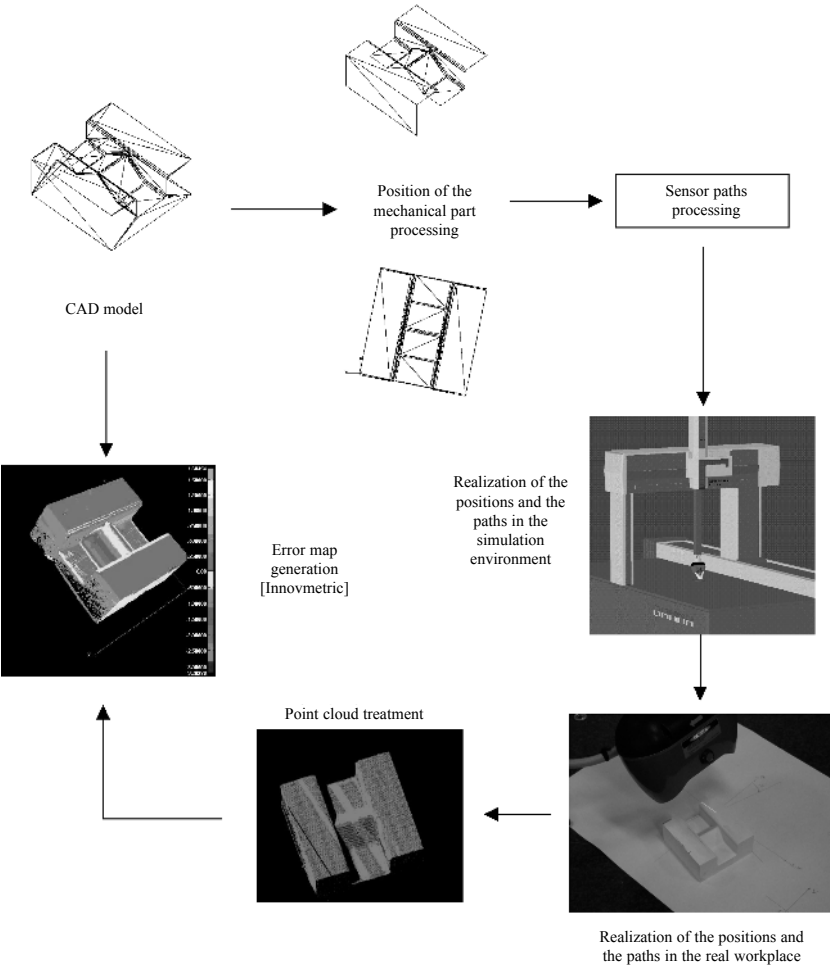


Fig. 9.40 Application

### 9.4.7 Conclusion

RP provides a rapid means of creating a physical object without too much fixturing if at all, while three-dimensional digitizing technology provides a means to capture geometric data from existing objects, albeit with possible defects. The translation of the point data clouds to complete CAD model or STL files allows a powerful combination for RE. It opens up vistas of possibilities to recreate existing objects especially for antique objects where blueprints no longer exist and also in recreating variants of existing objects. Repairs to replace existing objects can also be performed by taking cloud data of the good parts and mirroring to the damaged portion if a plane or planes of symmetry exist. The technologies are far from mature and much research work remains to be done in the hardware, software, and process planning.

### REFERENCES

- 1 Bernard, A., Davillerd, S., and Sidot, B. (1999) Analysis and automatic generation of 3D digitizing processes for laser sensor. In 8th European Conference on *Rapid Prototyping and Manufacturing*, Nottingham, UK, July.
- 2 Beaufile, P. (1999) Acquisition de données: tous les modes de capture sont bons. *Industries et techniques*, **806**, July.
- 3 Grimm, T. (1998) Reverse engineering applications for rapid prototyping. In Proceedings of *Reverse Engineering and Inspection Techniques/Applications*, Troy, Michigan, April.
- 4 Humbert, D., Cunin, D., and Barlier, C. (1997) Les différentes technologies de numérisation et capteurs associés. Actes 2èmes Congrès *Numérisation 3D*, Paris, France, May.
- 5 Sidot, B. (1997) Analyse préliminaire à l'automatisation du processus de numérisation 3D. *Mémoire probatoire CNAM*. Spécialité: Production Automatisée. Nancy, France, October.
- 6 De Baillenx, H. (1998) Etat de l'art des procédés de digitalisation. Forum commercial *Numérisation 3D*, Paris, France, May.
- 7 Bernard, A. and Taillandier, G. (1998) *Le Prototypage Rapide*. (Editions Hermès, Paris, France).
- 8 Bourdet, P., Lartigue, C., and Contré, A. (1998) Les capteurs 3D: état de l'art. Problématique liée à la précision des systèmes de numérisation 3D. Actes *Numérisation 3D/Human Modeling 98*, Paris, France, May.

- 9 Djeapragache (1998) Les capteurs 3D: état de l'art. Actes *Numérisation 3D/Human modeling 98*, Paris, France, May.
- 10 Beaufiles, P. (1995) La numérisation des formes tridimensionnelles. *Industries et Techniques*, **756**, January, 64–67.
- 11 Beaufiles, P. (1996) La rétroconception à votre portée. *Industries et Techniques*, **768**, February, 63–66.
- 12 Dubois, P. and Aoussat, A. (1998) Formalisation du processus en prototypage rapide: du besoin au choix des outils. Actes 7èmes Assises Européennes du *Prototypage Rapide*, Paris, France, November.
- 13 Kraus, K. and Waldhusl, P. (1998) *Manuel de photogrammétrie. Principe et procédés fondamentaux* (Edited by P. Grussenmeyer and O. Reis), (Editions Hermès, Paris, France).
- 14 Cadio, A. (1998) Exemples de réalisations utilisant la stéréovision. Actes *Numérisation 3D/Human modeling 98*, Paris, France, May.
- 15 Orteu, J.-J. (1998) Mesure de déformation 3D par stéréovision. Actes *Numérisation 3D/Human modeling 98*, Paris, France, May.
- 16 Ip, W. L. R. and Loftus, M. (1996) Shaded image machining: a reverse engineering approach using computer vision techniques and analytical methods to reproduce sculptured surfaces. *International Journal of Production Research*, **34**(7), 1895–1916.
- 17 Cadio, A. (1998) Calibrage d'un imageur laser 3D. Actes *Numérisation 3D/Human modeling 98*, Paris, France, May.
- 18 Dastarac, D. (1998) La tomographie X au service du prototypage rapide: numérisation et contrôle. Actes 7èmes Assises Européennes du *Prototypage Rapide*, 1998, Paris, France, November.
- 19 Ménégazzi, P., Henriot, S., Dupont, A., and Martineau, N. (1998) Utilisation de la tomographie scanner X pour le contrôle et l'analyse de pièces moteur complexes. Actes *Numérisation 3D/Human modeling 98*, Paris, France, May.
- 20 Rollins, J. and Jurick, G. J. (1998) Points to product and reverse engineering applications using X-ray scanning and computed tomography. In *Proceedings of Rapid Prototyping and Manufacturing '98 (RPM'98)*, Dearborn, Michigan, May, pp. 349–404.
- 21 Jurick, G. (1998) Reverse engineering for product development.

- In Proceedings of *Reverse Engineering and Inspection Techniques/Applications*, Troy, Michigan, April.
- 22 Neel, S. T. and Yancey, R. N. (1998) Engineering applications of high-energy computed tomography. In Proceedings of *Reverse Engineering and Inspection Techniques/Applications*, Troy, Michigan, April.
- 23 Griffin, A., McMillan, S., and Knox, C. (1998) Practical examples of reverse engineering using CT data. In Proceedings of *Reverse Engineering and Inspection Techniques/Applications*, Troy, Michigan, April.
- 24 Revue Gom News. *Gom* (1999) **1**, January.
- 25 Charnoz, F., Aeschel, E. H. N., and Maidhof, A. (1994) Digitalisation par voie optique. Nouvelle percée dans le monde du prototypage rapide. Actes 3èmes Assises Européennes du *Prototypage Rapide*, Paris, France, October.
- 26 Charnoz, F. (1996) La digitalisation par voie optique, applications industrielles. Actes 5èmes Assises Européennes du *Prototypage Rapide*, Paris, France, October.
- 27 Sakar, B. and Menq, C. H. (1991) Smooth surface approximation and reverse engineering. *Computer-Aided Design*, **23**(9), 623–628.
- 28 Turk, G. and Levoy, M. (1994) Zippered polygon meshes from range images. In Proceedings of *SIGGRAPH'94*, pp. 351–358.
- 29 Hoffman, R. and Jain, K. (1987) Segmentation and classification of range images. *IEEE Pattern Analysis and Machine Intelligence*, **9**(5), 608–620.
- 30 Varady T., Martin, R., and Cox, J. (1997) Reverse engineering of geometric models – an introduction. *Computer-Aided Design*, **29**(4), 255–268.
- 31 Chivate, P. N. and Jablowski, A. G. (1993) Solid-model generation from measured point data. *Computer-Aided Design*, **25**(9), 587–600.
- 32 Weir, D. J., Milroy, M. J., Bradley, C., and Vickers, G. W. (1996) Reverse engineering physical models employing wrap-around B-spline surfaces and quadrics. *Proceedings of the Institution of Mechanical Engineers, Part B*, **210**, 47–157.
- 33 Piegl, L. and Tiller, W. (1995) *The NURBS Book* (Springer-Verlag, Berlin).
- 34 <http://www.paraform.com>
- 35 Soucy, M. and Laurendeau, D. (1995) A general surface

- approach to the integration of a set of range views. *IEEE Pattern Analysis and Machine Intelligence*, **17**(4), 344–358.
- 36 Fang, T. P. and Piegl, L. (1995) Delaunay triangulation in three dimensions. *IEEE Computer Graphics and Applications*, **15**(5), 62–69.
  - 37 Cignoni, P., Montani, C., and Scopigno, R. (1998) DeWall. A fast divide and conquer Delaunay triangulation algorithm in  $E^d$ . *Computer-Aided Design*, **30**(5), 333–341.
  - 38 Lawson, C. L. (1997) *Software for  $C^1$  surface interpolation*. *Mathematical Software III* (Academic Press).
  - 39 Choi, B. K., Shin, H. Y., Yoon, Y. I., and Lee, J. W. (1988). Triangulation of scattered data in 3D space. *Computer-Aided Design*, **20**(5), 239–248.
  - 40 Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. (1992) Surface reconstruction from unorganized points. *Computer Graphics* (Proceedings of *SIGGRAPH*), pp. 71–78.
  - 41 Randrup, T. (1998) Approximation of surfaces by cylinder. *Computer-Aided Design*, **30**(10), 807–812.
  - 42 Jang, B. K. and Chin, R. T. (1993) Reconstructable parallel thinning. *International Journal of Pattern Recognition and Artificial Intelligence*, **7**(5), 1145–1179.
  - 43 Liu, G. H., Wong, Y. S., Zhang, Y. F. and Loh, H. T. (2002) Adaptive fairing of digitized data points with discrete curvature. *Computer-Aided Design* (in Press).
  - 44 Prieto, F. (1999) *Métrie assistée par ordinateur. Apport des capteurs 3D sans contact*. PhD thesis, INSA, Lyon, France.
  - 45 Prieto, F., Redarce, T., Boulanger, P., and Lepage, R. (1999) CAD-based range sensor placement for optimum 3D data acquisition. In *3DIM'99*, Ottawa, Canada, October, p. 128.
  - 46 Bernard, A. and Véron, M. (2000) Visibility theory applied to automatic control of 3D complex parts using plane laser sensors. *CIRP Annals*, **49**(1), 113–118.
  - 47 Chang, S. and Fengfeng, X. (1999) Model-based scanning path generation for inspection. In *3DIM'99*, Ottawa, Canada, October, p. 118.

## WEBSITES

- 1 [www.reverse-eng.com](http://www.reverse-eng.com)
- 2 [www.romer.fr](http://www.romer.fr)

- 3     *www.krypton.be*
- 4     *www.renishaw.com*
- 5     *www.sms\_ct.com*
- 6     *www.image3.com*
- 7     *www.materialise.be*
- 8     *tomoadour.fr*
- 9     *www.spincore.com/nmrinfo/nmr\_s.html*
- 10    *www.steinbichler.com*
- 11    *www.digibotics.com*
- 12    *www.metris.be*

*This page intentionally left blank*

# Chapter 10

## Virtual prototyping: a step before physical prototyping

**S Morvan and G Fadel**

Clemson University, Mechanical Engineering Department, Clemson,  
South Carolina, USA

### ABSTRACT

Free-form fabrication (FFF) or rapid prototyping (RP) machines transform objects drawn on a computer into tangible artifacts. Currently, this technology is weakened by the link between computers and FFF machines: the .STL file, which contains a series of triangles representing the skin of the object to be prototyped. A prototype, reflecting precisely the evolution of a concept within a design cycle and allowing a systematic inspection and verification, is essential. That prototype can be either a physical model or a virtual model, providing it allows this inspection/verification stage. A system addressing this need in a virtual environment was designed at Clemson University. IVECS, the Interactive Virtual Environment for the Correction of .STL files is a tool that was originally designed to allow minute surgery to be performed on faulty tessellated models. IVECS imports .STL files and detects tessellation errors before either automatically or manually repairing them. This chapter describes how IVECS is used not only for the correction of .STL files, but also for enabling virtual prototyping, and for educating the users of FFF technologies. It then describes the efforts made to expand the virtual prototyping aspect and ultimately to use the World Wide Web and Java to provide superior visualization to users from their own computer.



## 10.1 INTRODUCTION

Prototyping allows design engineers to embody their work, thus establishing a communication channel. Manufacturing engineers can consider the manufacturing issues of the part or its ease of assembly; they can give a better estimate of the cost and difficulty of making the artifact and possibly suggest alternatives or modifications. Alternatively, marketing experts can start to assess the success of the product by polling customers, asking them to consider ergonomic and aesthetic issues based on the prototypes. Currently, prototypes are generated using three different media: computer numerical control (CNC) machines and conventional manufacturing techniques, three-dimensional printers (rapid prototyping or free-form fabrication machines), and clay sculpting. All three produce a physical object that may or may not allow full functional evaluation, but that can be used to study the issues mentioned above. Each has its place in the design and manufacturing process. Both the CNC/standard manufacturing made prototype and the rapid prototyped part start as a rendering on the computer. CAD is the first step in this process, and the ability to render the three-dimensional physical part depends on the output of the CAD modeller and on the user of the technology. If that output is incorrect, contains flaws, or is incomplete, the computer-controlled machines may not operate and an additional design iteration will be required. Our contention is that by inserting a step between the CAD and the physical rendering, we can enable virtual prototyping and reduce if not eliminate the CAD-generated errors whether due to the CAD system translators or to the users. Additionally, the virtual prototype can provide sufficient feedback to shorten the whole design cycle by reducing the need for physical embodiments. This chapter concentrates on the computer-generated prototypes and describes the functionality of a virtual prototyping system.

## 10.2 VIRTUAL PROTOTYPING

Prototypes serve many purposes, among which is the visual feedback often used to assess model aesthetics. In such tests, the visual experience of the part takes priority over its mechanical strength or its fit within a mechanism. The purpose of such prototypes is to prove early in the design that the direction taken is aesthetically pleasing and possibly functionally correct to the targeted customers. Additionally, it can be used to assess manufacturing issues. Virtual prototyping aims at 'giving

life' to entities created within a CAD system. It allows human users to naturally experiment with objects resulting from a digital design process. The natural interaction is as simple and as elemental as walking around a car and observing the reflection of the surroundings on the body parts or examining a cellular phone with different screen lighting. Virtual prototyping provides further the means of studying a design regardless of its completeness. The life span of a virtual prototype is not as critical as its physical counterpart's. Recreating the virtual prototype is a question of hours or overnight computer jobs versus days, weeks, or months for a physical prototype. Changes, updates, and other last minute add-ons can be carried out more efficiently, without the fear of rendering obsolete a prototype eagerly awaited for.

Hinzmann (1) presented the benefits of virtually prototyping conceptual designs. Such a technique, when used earlier in the design process, unambiguously conveys information between designers. Because, as mentioned in reference (2), 'humans are spatial information processors', the use of videotapes featuring realistic images of a design guarantees a limpid information flow between teams of different specialties (aesthetics, production, marketing, etc.). The immediate benefit drawn from using the computer-generated images is the flexibility in the visualization of different configurations. For instance, colours can be altered, different textures can be applied, leather or cloth can cover the seats of a virtual car, walnut can be used for the dashboard and tinted glass can be utilized for the windshields, all at the cost of overnight jobs on computers.

Beside the time savings that can be expected from the use of computer-generated models, there are some additional benefits, among which is the prototype's flexibility and transportability. The flexibility of such a prototype is greater than that of its clay counterpart: the fear of rendering a model obsolete is no longer a hindrance in the design process since a modification is typically a question of hours. Transportability of digital prototypes is just another benefit: expensive secret set-ups are reduced to a conventional videotape, a CD, or at most, to a virtual reality (VR) set-up. Market research can take place quickly and more efficiently because of the added tailoring abilities.

Caterpillar used VR to evaluate the operator's field of view and the operational effectiveness of a future earth-moving machine. NASA used VR to visualize computational fluid dynamics data in a virtual

wind tunnel (3), helping engineers to extract critical information quickly from the 50 gigabytes file generated by the finite element solver. General Dynamics (4) used VR to analyse the maintainability of a new submarine, along with the University of Pennsylvania's computer avatar Jack application, which constrained the extraction/removal path to human-dependent parameters. Similarly, Boeing used Jack to do maintenance evaluation on the B777.

More recent work on the merging of CAD and VR has focused on an essential component of all VR applications, the interaction handler/manager. Whether it is by using natural interaction means such as speech and gestures in the COVIRDS System (5), or by allowing free-form surfaces to be manipulated and sculpted intuitively (6), or by designing rehabilitation aids in a virtual environment (7), the aim of these research efforts is not to match the capability and completeness of traditional CAD systems but to uncover what metaphor is best suited for a given CAD task in an environment surrounding the user. Other areas of concern are performance and design considerations for VR applications. Examples of these are a clever combination of various techniques for optimizing polygon bandwidth on three-dimensional workstations (8), and the considerations for the design of a persistent simulation (notion of state and temporal anteriority) environment for virtual manufacturing (9). It should be noted that unlike a conventional, graphical user interface (GUI)-based application that inherits a decade's worth of improvements and refinements in its underlying software model, a VR application does not share the same advantage. The use of a VR application is usually specific to a visualization or design task, making the sharing and generalization of software design issues a difficult task.

These users of virtual prototyping show that the technology is still in its infancy, and its potential is still to be explored. Our interest in it surfaced from our need to build accurate physical renderings of CAD data. Since this ability depends on the data format, which is used to bridge the computer world and the fabricator world, we discuss that data format next and the issues related to it.

### 10.3 THE .STL FILE FORMAT

The .STL file format is the *de facto* standard (10) used for data exchange between CAD systems and RP machines. It is a collection of

triangles (a tessellation) forming the boundary of the artifact to be prototyped. Initially designed in 1988, this file format was inherited from the internal representation of three-dimensional objects used in PHIGS [Programmer's Hierarchical Interactive Graphics System, (11)]. PHIGS was the most widely used library for mainframe-based CAD systems. It used triangle tessellation to display CAD models (three-dimensional objects) on the terminals' screen (two-dimensional devices). At that time, a tessellation of 20 000 triangles per object (12) was considered to be genuinely sufficient to appreciate a model's finest details, especially considering the capabilities of the computers. Today's CAD users tend to consider 20 000 triangles as a bare minimum for an accurate three-dimensional hardcopy (.STL files often contain more than 40 000 triangles). Thus, the adage 'if it renders OK, it will build OK' had to be revised and most CAD vendors felt the need to develop a separate function to generate effective .STL files. The user now constrains the maximum error between a surface and its faceted representation (triangles) to be smaller than a given value usually designated as the chordal tolerance. An .STL file defines a model as a set of triangles forming the material boundary. It specifies that an edge shall be shared by two and only two triangles and that each triangle obeys the right-hand rule (curling the fingers of the right hand along the edges of the triangle according to the order of the vertices should point the thumb towards the outward normal). Thus, the triangles should have exactly three adjacent triangles, one along each edge. Furthermore, an .STL model should be 'water-tight': the volume enclosed by the triangles is where the material will be located, and neither hole nor crack is allowed since this would introduce an ambiguity about the location of the matter. This is a necessary condition to generate a physically valid part.

There is a rather strict but often ignored distinction between a general-purpose CAD model and a CAD model used to build a prototype. While a generic model extends the designer's view of his/her concept, the model used to build the prototype extends the CAD system's understanding of the solid. In effect, by looking at an .STL file, it should be a simple task to resolve the matter's location, and thus to deduce what the object really looks like. Practically, this distinction is not as simple as it seems for several reasons: the .STL file generators can be applied to unclosed solids or partially described solids, and no provision for 'self-test' or intelligent interpretation of the CAD data is provided. Furthermore, there is reluctance from CAD users to

incorporate or understand rapid prototyping issues during model creation. For many, the generation of an .STL file is considered to be just another function of the CAD package such as printing the drawing. Moreover, the extensive collection of know-how provided in CAD packages for surfaces creation or finite element meshing falls dramatically short for the creation of RP parts, leaving the user no other choice but to learn RP models creation by trial and error.

In an .STL file several types of errors can be uncovered. Some are due to a weak tessellation algorithm in the CAD system, which results in either misoriented facets (e.g. the outward normal not consistent throughout the model) or non-closed shells (presence of gaps or holes). Others are due to the user's ignorance of .STL files' basics requirements (internal walls, self-intersecting geometry) or to the disregard of practical issues of file size and process constraints. The internal wall problem may not really hinder the building of the part, but it may cause problems to the finished prototype. In the case of the stereolithography process in which a laser beam hardens a resin, the areas with internal wall problems are polymerized twice, resulting in over-cure and possible warpage of surfaces.

The .STL file is the common denominator between most CAD systems and rapid prototyping machines. Furthermore, the ability to control the accuracy of the tessellation (by setting the chordal tolerance adequately, for instance) affects the quality of the approximation and fidelity of representation of a design, at the price of potentially *overly* tessellated models. However, if viewed as a collection of triangles in space, the .STL file can be put to use in virtual prototyping applications where the artifact can be coloured or painted to resemble a finished product. As reference (13) points out, there is a growing trend for mechanical CAD users to be able to give realism (read 'virtual prototyping') to digital designs. Previous work at Clemson used .STL files to allow walkthrough using a flat shading rendering algorithm (14), and the work presented here took this approach to a higher level, by allowing parts to be made transparent, chrome plated or smooth shaded, or even cut.

Before detailing the implementation, the next sections survey the various efforts at manipulating and correcting the .STL files. These approaches provided the impetus for the research presented.

## 10.4 PREVIOUS WORK

Most tessellation routines find their origin in the algorithms used by the CAD systems to compute solid views (e.g. smooth shaded views). CAD systems rely at some point on an unevaluated, surface-based representation of solids: roughly all solids are assembled in surfaces (the B-rep, or boundary representation). Each surface is mathematically defined by a function of two variables. These two variables are contained within a finite interval named the parameter space. When a surface is partially used within a CAD model (e.g. it is trimmed), the number of constraints on the parameters increases, leading to a more complicated parameter space (15). A subdivision algorithm transforms surfaces into triangles by recursively dividing surfaces. When the surfaces obtained are flat enough, they are replaced by triangles. The process of dividing surfaces is a trivial problem (16). In practice, surfaces are replaced by sculpted models, and the ordinary parameter space becomes increasingly complicated as surfaces get trimmed across most of their edges to form the desired geometry. Furthermore, the difficulty of correctly tessellating a model stems from boundary curves, which result from the intersection of two surfaces. A boundary curve belongs to two adjacent surfaces, and as such, is evaluated/sampled twice (once for each surface) during the tessellation. If the parameter space used to sample the curve differs slightly from one surface to the other, non-coincident vertices may emerge at the junction of the two surfaces.

Rock and Wozny (17) introduced the preliminary steps in correcting .STL files. The principle is that of inferring topology information from a 'bucket of facets' (e.g. the unordered list of triangles), which is achieved using efficient detection techniques and intuitive data structures. The paper presents its 'topology reconstruction concepts' as based on the merger of vertices, the creation of faces and edges, and by determining face and edge relations. The merger of vertices consists of unambiguously identifying vertices listed throughout the .STL file, and assigning them a unique entry pointing to a list of vertices. The unambiguous identification of all equivalent vertices is difficult to achieve because of the round-off errors; hence the requirement for a robust search/add algorithm to be used with a convenient data structure. The face and edge creation is straightforward from this point: a triangle and an edge are identified as a combination of, respectively, three and two integers that point to the list of vertices.

Then, the face and edge relationships are created. Determining the face relationships is done by searching faces that share an edge, and cross-referencing them; similarly the edge relationship is created (edge relationships are useful while slicing the part). The information gathered here (the indexed representation and the face relationships) clearly identifies errors in the file, and serves as a basis for many related papers.

Bøhn and Wozny (18) presented a simple yet efficient approach to the correction of errors within a faulty .STL file. An intuitive algorithm to correct triangle orientation is presented, which is based on an indexed representation of the facets, similarly to reference (17). The indexed representation is the first step for correcting .STL files; it consists of extracting and listing all the vertices forming the corners of the facets in a separate table. Triangles are now specified using pointers to the appropriate vertices. Using this, checking the orientation of triangles is straightforward: the neighbours of a triangle made from vertices #u, #v, and #w must contain an indirect permutation of either #u and #v, #v and #w, or #w and #u in the list of vertices. The iterative patching of triangles over holes (or punctures) with a 'lid-surface' is outlined. Once a puncture has been detected, a closed curve made from the free edges is identified and is filled with facets of harmonious proportions: if a small triangle is near a big triangle, their common diagonal is flipped to create two equally sized triangles.

Mäkelä and Dolenc (19) described a set of procedures for correcting errors in .STL files, based on an adaptive space subdivision data structure. Several methods to automatically correct errors are presented: triangles are oriented using graph theory and small gaps are filled. Filling gaps attempts to preserve the local smoothness by considering the relative orientation of triangles rather than their size (as did the previous algorithm). Finally, the authors conclude by raising a philosophical issue while correcting faulty files: the protection of the original intents of the designer. More importantly, the authors mention the difficulty of algorithmic-based approaches. They suggest what benefits could be drawn from an interaction between the user and the system.

More recently, Martin (20) proposed a public domain implementation of a correction suite for .STL files, ADMESH. The system performs surprisingly well on simple problems (misoriented normals and



disjoined edges), but the patching of a hole is a more problematic task that works only in some cases. Also, ADMESH allows parts to be scaled, translated, mirrored, and grouped, helping RP users to manipulate .STL models.

The research presented focused on the validation of robust algorithms that would automatically correct errors found in .STL files. However, little attempt was made to render .STL files on the screen of a computer, either for inspection or validation purposes, or better yet, to manually fix the part. IVECS was designed to provide these functionalities, and to some degree propose the assistance of the latter algorithms.

### **10.5 IVECS, THE INTERACTIVE VIRTUAL ENVIRONMENT FOR THE CORRECTION OF .STL FILES**

The primary goal for IVECS was to allow users to have their ‘hands on’ .STL models by proposing the visualization and the interactive correction of .STL files. IVECS detects errors and points them out to the user, who may choose to correct them. IVECS’ ability to automatically solve errors is limited to normal orientation problems and round-off errors. These two can be performed automatically to reduce the number of errors present throughout the file, so that the user is solicited solely for the patching of holes or the removal of extraneous facets (e.g. the internal wall problems). Essentially, IVECS supports the addition, the removal, and the reversal of facets. Additionally, the offset of facets is also proposed. IVECS supports both a GUI and a VR-based user interface, although the latter does not offer a comparable level of versatility.

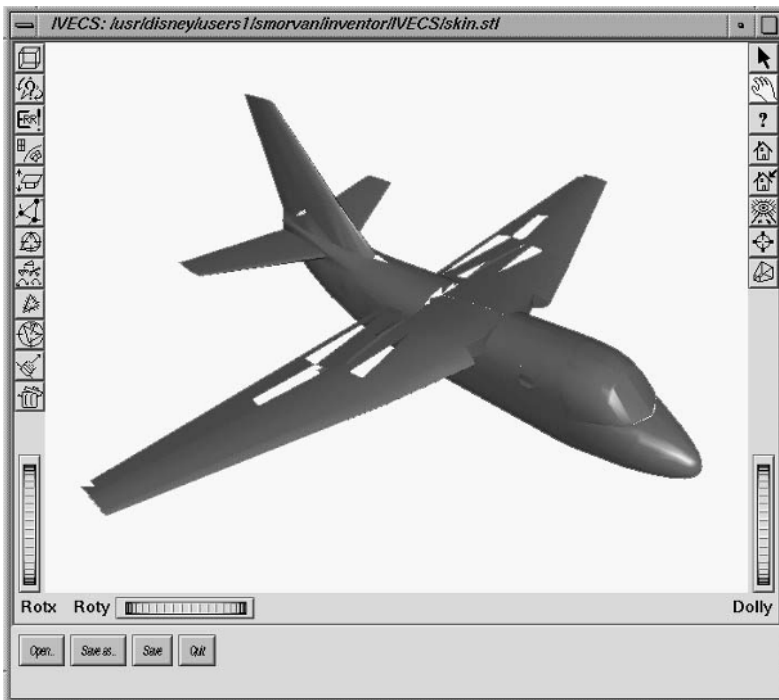
IVECS uses OpenInventor (21) for the rendering engine and the three-dimensional manipulation. The rendering can be done in a variety of ways: hidden lines for instance show the tessellation; flat shaded rendering shows the product as it will build on a rapid prototyping fabricator such as the SLA apparatus; and smoothed shading shows the model with rounded edges, comparable to a sanded prototype. The initial goals for this system were to support the reversal of a facet, its removal, or the patching of a crack. From this, and from geometries too complicated to visualize, several functions emerged. These functions can be separated into two groups:



1. *Mode functions.* These set the system in a particular type of visualization mode such as ‘slicing plane mode’ or ‘chrome-plated mode.’ Mode functions are not exclusive: a part can be set in chrome-plated mode and slicing plane mode at the same time.
2. *Facet functions.* These are the functions acting on facets. Some of these are reversal of the normals, removal of a facet, or creation of a set of contours.

A view of the overall user interface layout is presented in Fig. 10.1. On the left side of the rendered view is the toolbar. The upper half of the toolbar is dedicated to mode functions while the lower half is dedicated to facet functions. Most functions depend on parameters that can be set by holding the shift key on the keyboard while selecting the corresponding button. The upper and lower halves of the toolbar are separated by the ‘selection of facets’ tool.

The ‘selection of facet’ tool allows facets to be selected. When this function is activated, the cursor changes from an arrow to a cross hair;



**Fig. 10.1** The user interface layout

and a facet gets highlighted in red (e.g. selected) whenever it is clicked. Along the principles of standard graphical user interfaces, the multiple selection of facets is possible: when the shift key is held at the time the selection occurs, the facet is added to the list of selected facets. A facet gets deselected if it is selected twice, and all facets are deselected if the mouse is positioned and activated on the empty space surrounding the artifact. The action of selecting the facets prepares the input for the facet functions. A set of facets can be selected and the mode functions can be accessed without interfering with the current selection.

### **10.5.1 Mode functions**

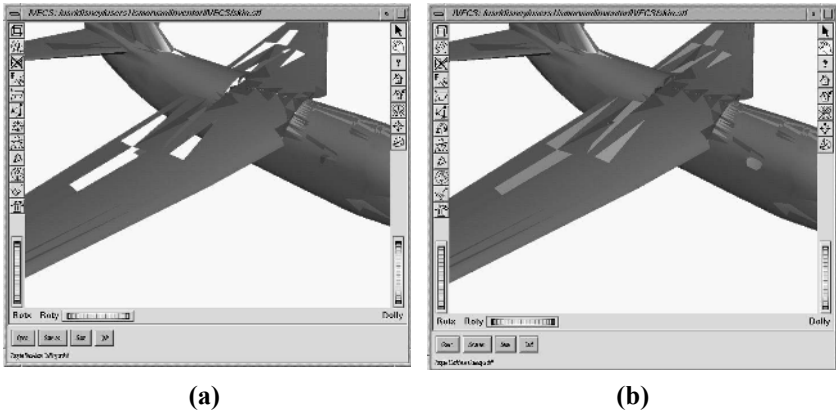
The mode functions consist of: transparent, backface culling, errors, chrome, vertex, and slicing plane. Toggling one mode on/off usually results in visible changes in the display area (the artifact is rendered differently) and the corresponding button is set in a 'toggled' state.

#### ***10.5.1.1 Transparent mode***

The part is rendered as if it is translucent to light, revealing internal details. The rendering of transparent parts uses an alpha-additive blending process, resulting in a greater opacity where many facets are aligned with the direction of sight. This feature is specifically computer intensive, and on-board acceleration greatly enhances the interactive experience.

#### ***10.5.1.2 Backface/misoriented facet mode***

The responsiveness of the system is greatly improved if only one side of a facet is rendered, thus relieving the rendering load from about half of the total number of facets present in the viewing frustum. To achieve this, an assumption is made on which side to render: the system computes the outward normal of each facet and derives its visibility from the current point of view. When backface culling is on, the misoriented facets are not rendered correctly (visually behaving like holes in the skin of the artifact), possibly misleading the user. From this ability to select which side of a facet to render, a slight modification eases the visualization of misoriented facets: by applying distinct materials (e.g. material specularities: reflection, ambient colour) to both sides of every facet, misoriented facets can be spotted easily (Figs 10.2a and b). In this particular rendition, the light colour triangles are misoriented.



**Fig. 10.2 The display of misoriented facets**

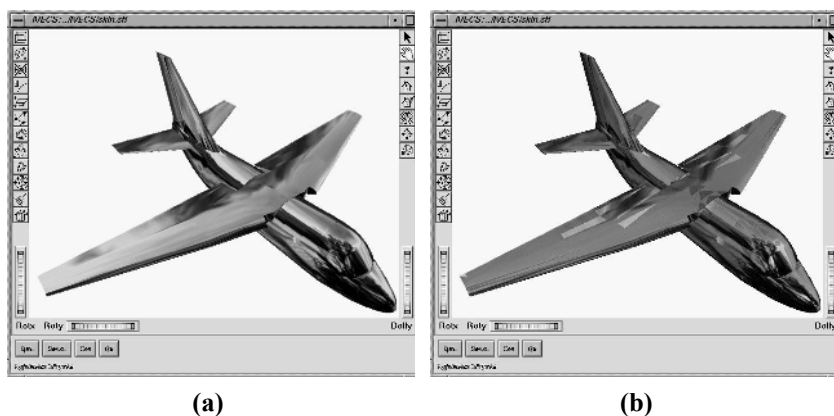
#### **10.5.1.3 Errors mode**

The errors mode displays the errors found in the .STL file. The errors are displayed by colouring problematic facets with a shade differentiable from the overall colour of the part. For instance, facets missing one neighbour are coloured in pink, and those having an extraneous facet are coloured in yellow. This visual cue is essential in the quest to virtually prototype. The human eye is trained to look for differences, and can easily spot changes in colour. In addition to the visual response, the number of erroneous triangles along with the classification of the errors is presented to the user on a separate screen.

#### **10.5.1.4 Chrome-plated mode**

This mode makes the part look chrome plated. A texture is placed inside a sphere, and the reflection of that texture on the part is computed. Thus, whenever the part is moved, the reflection is recomputed. Though this dynamic texturing scheme is computationally intensive, it results in an increased understanding of the geometry presented to the user. The principles that make this technique so efficient at grasping most of the geometric features at a single glance are based on surface interrogation techniques. Surface interrogation techniques (22) are used by car-makers to match a stylist's freehand sketch with the machined prototype that emanated from it. Briefly, the technique consists of observing the reflection of neon lights on the polished surface. The increased amount of information received from the reflection of a known pattern on a

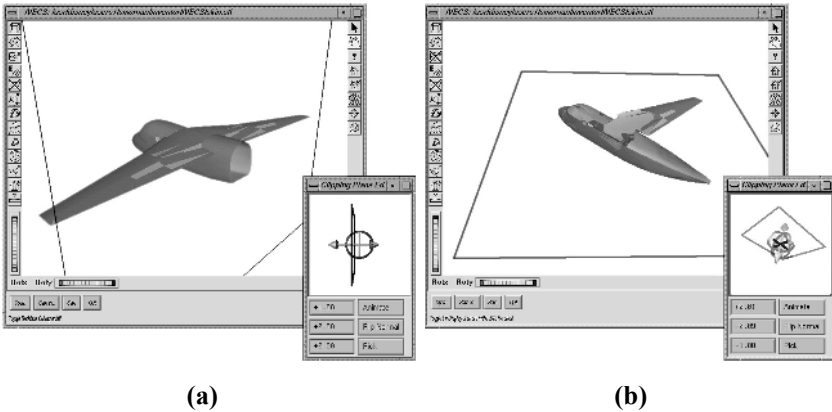
polished surface results in an intuitive, natural way of assessing the curvature or smoothness of a surface. The chrome effect promotes the same type of natural experience with the user, and unexpected waviness resulting from a poor surface management at the CAD level is easily flagged. Several types of textures can be used: a scenic view consisting of flowers, sand and a blue sky, and red stripes on a white background have been used to date. In mechanical design, entirely chrome-plated parts are rare (especially at the prototyping stage) and lacquered ones are more common. Noting this, the experience can be enhanced if the original colour of the model is retained underneath the chrome plating, akin to a glossy surface finish, which can be attained using alpha blending of the textures with the original colour of the part (Figs 10.3a and b).



**Fig. 10.3 Different chrome-plating modes. Note the persistence of individual facets' colouration**

### 10.5.1.5 Clipping plane

As an alternative to the 'transparent mode' for visualizing the inside of a part, the clipping plane mode features a moving plane that intersects the artifact and hides the geometry lying in its negative semi-space. The positive semi-space is defined by the normal of this plane. This plane can be conveniently oriented and translated (via a three-dimensional graphic manipulator in a secondary window, Figs 10.4a and b), allowing artifacts to exhibit their interiors. Also, this mode intuitively simulates the building process: the visual reconstruction from bottom to top helps to assess the best build direction and



**Fig. 10.4 The slicing plane**

eventually confirm the results obtained later from a post-processor of .STL files that determines optimal build orientation (23).

#### **10.5.1.6 Facet functions**

The facet functions are the heart of this system: they allow the modification of the artifact in several ways. Facets can be added, deleted, offset, and reversed, and colour-coded contours schemes can be produced. Any of these functions affects only the selection, resulting in changes in the internal data structures. No undo function is provided, since an error can be easily recovered by either recreating a facet (an accidental deletion), removing it (for the erroneous creation of a facet), or 're-reversing' it (for the accidental reversion of a correctly oriented facet). They often result in minute changes in the geometry, as opposed to the immediately observable changes produced by the mode functions.

#### **10.5.1.7 Creation of coloured contours**

To help visualize problems, sets of contiguous triangles are displayed in different colours. The neighbours (sharing one vertex) of a specific triangle are coloured in red, the neighbours ('subneighbours') of these neighbours are coloured in green, the next ones in blue, and so on, repeating red, green, and blue patterns concentrically around the selected triangle (the red, green, and blue colours choice is arbitrary; these could be set to virtually any colour not confusing the user). Whenever a triangle is flagged as problematic, its insertion in the

current set of subneighbours is delayed to the next set of subneighbours. Thus, problematic facets break the colour sequence: instead of, say, being coloured in green they are coloured in blue and the skipped colour immediately flags the error and its effect on the arrangement of vertices. Also, non-intersecting solids can be flagged easily: since these isolated artifacts do not share common vertices, the contours do not *spread* from one artifact to the other, resulting in the colouring of one of the artifacts while the other one is left untouched. Additionally, the coloured contours can be transformed to an active selection, equivalent to selecting all the coloured facets, which can then be deleted all at once.

#### **10.5.1.8 Patching of a hole**

To patch a hole or a crack, the user selects a pair of problematic triangles sharing one vertex and the system builds a triangular patch over the hole between the two triangles. If one (or both) of the triangles selected has more than one missing neighbour, several patches could be computed. In that case, the user browses through the set of possible solutions by repeatedly clicking on the mouse. The process is repeated until the hole is filled. If the 'errors' mode is on, the set of errors is updated each time a new facet is created: pink facets 'move' around the remaining holes, and when the surface finally becomes closed, the erroneous pink triangles totally disappear. This dynamic update guides the user towards achieving shell closure by drawing attention on remaining problems or newly created ones.

### **10.5.2 Other efforts**

#### **10.5.2.1 Virtual reality interface**

The goal of this work was to pre-process .STL files and supplement the FFF machine by visual prototyping. A natural way to inspect three-dimensional objects, and to circumvent ordinary desktop metaphor limitations, is to use virtual reality (VR). Building on a VR-based visualization of .STL files developed earlier at Clemson, portions of IVECS were implemented in virtual reality. The following sections will briefly describe the navigation techniques used (24).

Several interaction/navigation means were inspired from past work on VR navigation metaphors (25–27): the user had to be able to either 'grab' the part, enlarge it to view details, or navigate around it. The default state of the virtual environment (VE) application is named

‘flying mode’, the state where the user holds the part ‘grab mode’, and finally, when vertices are edited, ‘edit mode’. The transition between these states is based on the user’s intention as perceived by the system. That is, if the user selects a triangle, he/she is inspecting the arrangement of vertices at these points, and may eventually want to modify the object. Thus the ‘flying mode’ will automatically change to ‘edit mode’ if the user fires a ray at a triangle. If the ray is fired in the emptiness, the ‘grab mode’ becomes the new VE. A ray is fired by pressing a button on the three-dimensional mouse and aiming a transparent cone to a target; when the button is released, the intersection of this ray with the .STL object is computed by the system.

#### ***10.5.2.2 Flying mode***

The use of the three-dimensional mouse as a ‘thruster’ providing unidirectional motion was quickly abandoned, primarily because of its non-reversibility: whenever the user realized he/she was past a point of interest, he/she had to release the button pressed, and attempt to start a motion in the reverse direction (by pressing another button). Since the accurate orientation of the three-dimensional mouse in the VE was difficult to resolve most of the time, the direction of motion is derived (and kept fixed as long as the user holds the button down) from the head-mounted display’s (HMD) direction of sight (the user always looks in the direction in which he/she wants to go) and the three-dimensional mouse was used as a throttle to specify the speed of displacement. Thus when the user passes a point of interest, he/she could easily achieve back-up motion by reversing speed, measured from the variation in the three-dimensional mouse’s pitch angle since the motion was initiated. A hand can rotate along its roll axis roughly from  $-45^{\circ}$  to  $+135^{\circ}$ ; thus a range of speeds was computed to correspond to the rotational angle and provide fast back-ups and precise forward motions. By holding another of the three-dimensional mouse buttons the user can enlarge/reduce the part (in a fashion similar to the speed setting). Large zoom factors, quickly reached, help locate/visualize errors in the .STL file. As the limited field of view of the HMD failed to convey true depth cues (such as the relative size of objects), which are typically offset by stereoscopic vision, the VR lab’s floor was included in the VE at a fixed position under the tracking system. Thus whenever the user is walking, he/she walks on the floor, whereas when he/she is ‘flying’, a global translation is applied both to him/her and the floor (one would say that the whole room containing

the VR lab is moved). The part's scale can now be perceived in a more natural way: the visual occlusion by the floor helps to size and position the object in the VE.

### ***10.5.2.3 Grab mode***

As the user evolves in the VE he/she may also want to 'hold' the part in his/her hand. At the press of a button, an animation sequence is initiated and the VE application's state changes to 'grab mode.' Similarly to real-world situations, the virtual world (VW) must provide continuity and allow fixed points to remain relatively steady, the steadiness of these points being key to human orientation cues. The animation sequences smooth transitions between large geometric transformations occurring in the VW. Once the part is attached to the three-dimensional mouse (the user's hand) the part can be made closer/farther from the three-dimensional mouse, or it can be incrementally rotated along the roll axis of the user's hand, allowing quick access to all geometric features. When the part is released, another animation sequence is initiated to return the part to its original location.

### ***10.5.2.4 Edit mode***

The user, now familiar with the part, may wish to inspect the part and its enforcement of the .STL file format specifications. The .STL file describes an object as a closed skin/surface (or 'watertight' object) and as a succession of oriented (following the right-handed rule) triangle sets. Each triangle's edge is shared by two and only two triangles (that is, one triangle has only three adjoining triangles). The selection of facets is achieved as follows: the mouse is used to orient a finite ray (used as the axis of the selection cone) in the VE, and to select facets lying inside the selection cone. To fit the desired facets under the selection cone, the user has the choice to reduce/augment the base radius of the cone, broadening its selection accordingly. Raycasting is then used to compute the intersection of the ray with the facets.

### ***10.5.2.5 Virtual reality interface limitations***

The impact of the system was somewhat lessened by the VR hardware's limitations. Perhaps the most significant of these being the fluctuations of the tracking system, which become increasingly noticeable as the user gets close to details. At close range, the jittering becomes almost unbearable: the magnitude of the fluctuations is



amplified by the zoom factor obtained from the projection algorithm. Among other problems was the lack of real world/VW uniform mapping: the three-dimensional mouse (e.g. the user's hand) did not realistically appear attached at the tip of the user's forearms in the VE; it was either too far or too close from the user's point of view.

It is anticipated that the use of a filter rejecting fluctuations in the 50–60 Hz range would greatly eliminate the jittering problem (28) and that a careful calibration of the VW would certainly help to resolve the orthotropic effects observed. The most essential element of this VR interface is this ability to interact with the users while they were in the VW, and to accommodate their need to switch between different modes. The selection of transitions between modes should have been at the user's discretion as opposed to the system's perception of user intent, which was disorienting most of the time.

#### ***10.5.2.6 World Wide Web based inspection of .STL files***

The core research stage of this project took place in 1994–1995, at a time when robust three-dimensional performance and strong development capabilities meant Unix-based workstations such as the SGI Indy or the Onyx RealityEngine. As the software evolved, a growing number of RP users (from a predominately PC-based environment) manifested their interest in IVECS. A decision to make IVECS available on PCs running on the Windows operating system was made. The port of IVECS to the Windows operating system was successfully completed, and initial testing shows no significant performance trade-offs for comparable (entry-level) workstations.

There is still a wide base of users operating on computers without three-dimensional support, and in order to provide these users with the resources available at Clemson University, an attempt to bridge IVECS with the World Wide Web (WWW) was made. Based on a Java applet and software running on a host SGI machine, a rendering daemon monitors an incoming ftp directory. When a new .STL file is detected, IVECS is used to generate a collection of snapshots of different views of the model, along with a complete status report of the error(s), if any present, throughout the .STL file. The views of the part are such that they cover most of the angles under which the part can be examined; any given snapshot differs by a small increment in either the horizontal or vertical direction. A Java applet is then used to

map the displacement of the mouse in a window with the corresponding snapshot, giving the illusion of three-dimensional interaction. A good illusion of three-dimensional rendering is achieved for angle increments of  $10^\circ$  in both horizontal and vertical directions, for a total of 200 images, requiring about 1 Mb of 24 bits,  $320 \times 240$  pixels JPEG encoded images to be downloaded (thus minimizing download time). It is expected in the near future to give more control over how the images are rendered (e.g. image size, IVECS' mode, antialiasing, etc.) in order for people with fast internet connections to enjoy better resolutions.

## 10.6 CONCLUSION

The need to virtually prototype a product before actually creating a physical model or the finished part is critical to the shortening of the design cycle, which is dictated by global competitiveness pressures. We describe in this chapter an approach to virtually prototyping parts that have been designed on a CAD workstation and that are in the process of being used to build a physical prototype. The approach uses the .STL file format, which is a *de facto* standard for free-form fabrication processes. The functionality of the developed interface is described, along with its extensions to immersive environments and web interfaces. The use of the virtual environment provides users with superior visualization capabilities that far exceed those of current CAD systems. Furthermore, the use of such a tool has an even greater educational value and can contribute to making the designer more aware of manufacturing issues.

## REFERENCES

- 1 Hinzmann, B. (1995) Does rapid prototyping make sense for industrial design? In Proceedings of the 2nd Annual Eugene C. Gwaltney Manufacturing Symposium, Georgia Institute of Technology, Atlanta, Georgia, pp. 174–203.
- 2 Austakalnis, S. and Blatner, D. (1996) *Silicon Mirage, The Art and Science of Virtual Reality* (PeachPit Press).
- 3 Bryson, S. (1993) The virtual windtunnel: a high-performance virtual reality application. In Proceedings of *VR AIS 1993*, pp. 20–25.
- 4 Larson-Moyal, J. (1994) VR at work today. *Computer-Aided Engineering*, **64**, 68–70.

- 5 Dani, T., Chu, C.-C., and Gadh, R. (1997) COVIRDS: shape modeling in a virtual reality environment. In Proceedings of ASME Design Engineering Technical Conferences, Sacramento, California, paper DETC97/CIE4302.
- 6 Furlong, T. (1997) Virtual reality sculpture using free-form surface deformation. In Proceedings of ASME Design Engineering Technical Conferences, Sacramento, California, paper DETC97/DFM4511.
- 7 Krovi, V., Kumar, V., Ananthasuresh, G., and Vezien, J.-M. (1997) Design and virtual prototyping of rehabilitation aids. In Proceedings of ASME Design Engineering Technical Conferences, Sacramento, California, paper DETC97/DFM 4361.
- 8 Varshney, A., El-Sana, J., Evans, F., Darsa, L., Costa, B., and Skiena, S. (1997) Enabling virtual reality for large-scale mechanical CAD datasets. In Proceedings of ASME Design Engineering Technical Conferences, Sacramento, California, paper DETC97/DFM4371.
- 9 Lee, D. and Hahn, T. (1997) Generic modular operations for virtual manufacturing process engineering. In Proceedings of ASME Design Engineering Technical Conferences, Sacramento, California, paper DETC97/CIE4309.
- 10 Fadel, G.M. and Kirschman, C. (1996) Accuracy issues in CAD to RP translations. *Rapid Prototyping Journal*, **2.2**, 4–17.
- 11 Shuey, D., Bailey, D., and Morrissey, T. (1986) PHICS: a standard dynamic, interactive graphic interface. *IEEE Computer Graphics and Applications*, **6.8**, 50–57.
- 12 Deering, M. (1993) Virtual reality: where do all the triangles go? *IEEE Computer and Graphics*, **17**, 357–363.
- 13 Rowell, A. (1997) CAD/CAM special report: prototyping in a digital world. *Computer Graphics World*, **20.9**, September, 55–62.
- 14 Fadel, G., Crane, D., Dooley, L., and Geist, R. (1995) A link between virtual and physical prototyping. In Proceedings of the SME Rapid Prototyping and Manufacturing Conference, Detroit, Michigan.
- 15 Toriya, H. and Chiyokura, H. (Eds) (1993) *3D CAD Principles and Applications* (Springer-Verlag).
- 16 Sheng, X. and Tucholke, U. (1991) On triangulating surface model for SLA. In Second International Conference on *Rapid*

*Prototyping*, Dayton, Ohio, pp. 236–239.

- 17 Rock, S. and Wozny, M. (1992) Generating topological information from a ‘bucket of facets’. In Solid Freeform Fabrication Symposium, Austin, Texas, pp. 251–259.
- 18 Bøhn, J. and Wozny, M. (1992) Automatic CAD-model repair: shell-closure. In Solid Freeform Fabrication Symposium, Austin, Texas, pp. 86–94.
- 19 Mäkelä, I. and Dolenc, A. (1993) Some efficient procedures for correcting triangulated models. In Solid Freeform Fabrication Symposium, Austin, Texas, pp. 126–135.
- 20 Martin, A. D. (1997) Homepage for ADMESH, <http://www.varlog.com/products/admesh/>
- 21 Wernecke, J. (1994) *The Inventor Mentor* (Addison-Wesley).
- 22 Lennings, A. F., Peters, J. C., and Vergeest, J. S. M. (1995) An efficient integration of algorithm to evaluate the quality of freeform surfaces. *Computer and Graphics*, **19.6**, 861–872.
- 23 Kirschman, J., Kirschman, C., Fadel, G., and Greenstein, J. (1997) The Clemson Intelligent Design Environment for Stereolithography – CIDES 2.0. Solid Freeform Fabrication Symposium, Austin, Texas, pp. 569–576.
- 24 Morvan, S. and Fadel, G. (1996) IVECS: an interactive virtual environment for the correction of .STL files. In Proceedings of ASME Design Engineering Technical Conferences, Irvine, California, paper DETC96/DFM1305.
- 25 Ware, C. and Osborne, S. (1990) Exploration and virtual camera control in virtual three dimensional environments. In Proceedings of *SIGGRAPH*, pp. 175–183.
- 26 Liang, J. and Green, M. (1994) JDCAD: a highly interactive 3D modelling system. *Computer and Graphics*, **18.4**, 499–506.
- 27 Yoshimura, T., Nakamura, Y., and Sugiura, M. (1994) 3D direct manipulation interface: development of the Zashiki-Warashi System. *Computer and Graphics*, **18.2**, 201–207.
- 28 Nixon, M. (1998) The effects of metals and interfering fields on electromagnetic trackers. *Presence, Teleoperators and Virtual Environments*, **7.2**, April, 204–218.

*This page intentionally left blank*

# Chapter 11

## Open issues

**I Gibson**

Department of Mechanical Engineering, The University of Hong Kong

### **ABSTRACT**

A number of issues relating to software systems in rapid prototyping have been discussed in the previous chapters. In each of these chapters, many discussion points have been raised. Often, these points have been inconclusive, or open-ended. This is perhaps because the problem is too new, too difficult, not considered sufficiently important, or perhaps not fully understood at present. This chapter will collect some of the thoughts of each of the previous authors together. These are open issues that are important to general research in this area. It is hoped that the reader can take note of these and incorporate them into his/her own future research plans.

### **11.1 PROCESS CONSTRAINTS**

Many software researchers consider rapid prototyping (RP) as a collective term that represents a general technology that can be served by a general software algorithm. However, it is quite clear that when we look at any specific system, it will have advantages and limitations when compared with others. A user of selective laser sintering (SLS) technology casually stacks parts on top of each other when setting up a build. Users of fused deposition modelling (FDM) and stereolithography (SLA) would seriously consider the necessity of doing this before attempting it. Conversely, SLS has problems relating to thermal distortion and powder packing densities that require careful attention to complex geometry or large parts: factors that may have no singular importance to some other systems.

In fact, while generally considered to be geometrically insensitive, nearly all RP machines have problems associated with one kind of geometry or another. This may be thick wall, thin wall, or the transition between the two. It may relate to feature size, layer thickness, or other accuracy-dependent issues. What this means is that RP software must be tailored to suit the systems that it is designed for.

Consider, for example, the packing of parts on an RP machine build platform. Many software systems look at this as a classical packing problem based on geometrical analysis. This is a very naïve approach that has little or no benefit to the RP machine user. Parts that are placed too close to each other become difficult to separate. Orientation of parts often has other limiting constraints like accuracy or surface finish. These factors are somewhat dependent on the process used and also are not recognized during a purely geometric analysis.

Those software systems that do take process constraints (like surface finish) into consideration are currently not intelligent enough to deal with conflicting constraints. It may be possible to orient a part in order to give it the best surface finish from the point of view of layer thickness. This may however require the part to be oriented in such a way as to make it difficult to clean from the point of view of support structures. How does a software system come up with an appropriate compromise? This, and similar questions, certainly require the attention of software system researchers.

## **11.2 GENERALIZATION VERSUS SPECIALIZATION**

The above problem of process constraints makes it difficult to see commercial benefit in developing software for RP systems. If there are only 1500 FDM machines worldwide, how is it possible to justify building a software system suitable for such a small number?

There are however approximately 10 000 of all different kinds of RP machines and, even though this is still a small number, developing a software package that is suitable for all of them does start to make commercial sense. If we then go on to look at all the potential users of these machines (i.e. more than one software package per machine, or perhaps even Internet software systems) and develop a software system that can accommodate all their needs, then it starts to move into the realms of possible profitability.

Taking into account the steady growth of this market indicates that researchers and developers should start to seriously consider how software systems of a general nature could be created for RP technology. With the increasing importance of the Internet in everyday business, it is likely this software should be built around the Internet.

However, such software systems must also be usable. Simulation software should be able to distinguish between different systems, both with and without supports. Quoting software must be able to realistically compare different technologies without bias or ambiguity. This means that specialization must even be built into the general software. Such software maybe doesn't have to be so precise as to model thermal behaviour for example, but there must be sufficient knowledge of each system being covered to make it useful. Examples of this may include:

- Does the system use support structures?
- How many different types of material are available?
- Are there different build styles available?
- What would be the potential effects of changing position/orientation of the part?
- How does the system deal with batches?
- Etc.

### **11.3 DESIGN CONSTRAINTS (USER INTERFACE)**

It is therefore quite clear that many of the difficulties pointed out by the authors of the previous chapters of this book are constrained by part design or part specification matched against process constraints. The proposed solutions are generally tried against the experiences gained from using a small number of machines. In the previous section we noted that it might be difficult to scale or generalize these views to accommodate a wider range of technology and yet still make it useful.

There is a fine line between developing software so that it is easy to use and simplifying the system to the extent that it is useless or misleading. Taking too much control away from the user can result in a software system that can only provide a partial solution. If a simulation package makes the FDM process look similar to three-dimensional printing (3DP) or SLA without showing the differences in



speed, accuracy, or material properties, then it is not going to be particularly useful. Similarly, if the user (or customer) is unable to see the different results that can be had from adjusting the parameters of a specific machine, then this can only show a limited capability.

All this must be done without making the software unwieldy or complex. There is a need to provide guidance through the software, so that the experience gained from one user can be used as a back-up for others.

## **11.4 DATABASE MAINTENANCE AND VALIDATION**

A number of chapters have concentrated on the sharing of information between agents or systems. This has mainly been discussed in the form of archiving machine specifications along with materials information, process parameters, and other quantifiable data. However, it has also been considered that more intangible issues, like opinions and experiences, may also be useful for some of the software systems in development. There are two particular issues of paramount importance when considering such a database:

- How is it possible to collect this data together and keep it up to date?
- How to ensure that the data contained within the database is valid?

Keeping a database up to date means both adding new data and replacing data that have been superseded by more recent information or developments. If a number of databases are used at different sites, then there is a possibility of conflicting information in each version. The validity of the data very much depends on the data source. Users will have different opinions from each other and, more specifically, from the vendors. A vendor in particular does not want critical viewpoints incorporated into the database. Certainly at the least it has to have been validated by an accepted or trusted and independent source.

Even from the point of view of machine performance, validation is a difficult issue. Numerous benchmarks have been created to illustrate the capabilities of different RP technologies. Every benchmark has been criticized in one way or another. As soon as a new machine, material, or technique becomes available, the benchmark data become outdated and difficult to interpret. Also, very few of these benchmarks seem to incorporate error bounds or confidence factors.

When it comes to determining value for money, or the best way to look after a machine, or what methods should be used for casting or tooling, we are moving into the realms of qualitative viewpoints. This is very difficult to validate. If there is sufficient logic in the approach, it is very possible to come up with multiple approaches or solutions. However, it is highly unlikely that any database can have all these variables to hand. Any such database must therefore have a caveat attached to it.

## **11.5 STANDARDS**

One possible, at least partial, solution to the dilemma above is the use of a standards authority. Probably, the current RP market is of little concern to the standards authorities of the world, except where there are wider implications like materials or energy sources. Performing individual tests on RP technologies must lie in the hands of the users and vendors for the present and therefore research into testing standards (benchmarking) and validation techniques would be a very worthwhile direction to take for a market of such promise.

There is another issue relating to standards that has been raised by a number of authors in this book: that of data interchange. It is clear that developments in RP are likely to result in the demise of the standard STL file. The triangle-tessellated format used for model data is just too simplistic for systems that are aiming at combined or integrated processes. The latest colour RP systems are already finding it difficult to operate with STL. Multiple material systems will probably find it impossible. It is therefore essential that when a new standard is adopted, it must fulfil all the requirements of new RP technologies. It must be intelligently thought out and adopted by all parties concerned.

Since many new software systems will be based on system specifications as well as model data, it is important that standards are adopted for data interchange of machine data as well. If every new decision support system (DSS) or simulation package is built from scratch then progress will be slow and it will be difficult to incorporate new machines as they become available. Standards for specification of RP technology should also be seriously considered. In that way, when it is known for example how fast a machine can build, it is possible to compare it easily with others. At the moment, machines are quoted in terms of scan speed, layer times, numbers of layers per hour, volume produced per hour, etc. These figures are generally vague and related

to internal benchmark pieces. Because machine build times are so variable, it is easy to disguise just how fast (or slow) a machine actually is. Standards need to be established and adhered to.

## **11.6 DATA EXCHANGE**

It is widely known that exchanging data between non-similar CAD systems is both difficult and unreliable. In some ways this is inevitable since storage of model data is implicitly linked with the CAD kernel and modelling technique. Different CAD systems seek to gain advantage over competition by employing different kernels and techniques. By converting the model data into a format compatible with other systems, much of the modelling process is bound to be lost. One can imagine that this is going to be the same with any complex file arrangement and any application that is not dominated by one particular system or software approach.

Another issue regarding data exchange is time lag. As soon as a data exchange format is decided upon, this imposes a series of constraints that may affect future development. In an area that is rapidly changing, like RP, it is difficult to establish a data exchange standard. As mentioned above, the STL standard is starting to make it difficult to develop RP technology, particularly that being developed to support multiple materials. It is also important to consider database formats that permit the development of user-support systems described in chapters of this book. With such systems there is often a need to match machine capability with model data. The model data for current commercial systems may be in STL format, but the definitions concerning machine capability are not standardized and therefore it is difficult to establish a common framework that can permit comparison. Chapters on benchmarking, adaptive slicing, and decision support all allude to this problem. It is likely that there is a need for a concerted effort to benchmark systems in order to define this common framework. Such an approach must gain the support of the major system manufacturers in order to be successful.

## **11.7 DESIGN OPPORTUNITIES**

Rapid prototyping provides opportunities for designers to create parts that are difficult to produce using other methods. Until recently, this has been something of a novelty rather than considered of being of real

benefit. Now, RP machines have been demonstrated as viable for direct manufacture for certain types of product. Multiple material capabilities will also add a degree of functionality to parts that will continue to open up this market sector. The use of sacrificial regions (e.g. leaving unsintered powder around components in order to provide clearances) makes it possible to remove the need for assembly of some components and thus reduce the component count. Fully enclosing objects inside others is another ability of RP that can be exploited. Add to this the ability to grade materials, thus providing a mechanism for determining the properties in different regions of a monolithic structure, and the designer is provided with a whole new range of solutions.

This also presents the designer of the software systems with a whole new range of problems. Functionality becomes a driving factor in design to the extent that for certain types of product there is a need to specify the capability in terms of physical attributes other than shape. In other words, you may design the part in terms of strength or stiffness that is required. The CAD system will therefore have to give you both the shape and the internal material composition. Alternatively, it may be possible to seed the system with a maximum external shape. Optimization of that shape can then be performed provided the material properties of the multiple material systems are well understood and incorporated into the design algorithm. This implies there is a meshing between the geometric modelling system and the material modelling system.

A further issue related to this is the ability to trade one process off against another. Often an RP process may result in part distortion in some way for a particular design. If it is known how the distortion is going to embody itself, then there is an opportunity to develop a system that can compensate for it. For example, a flat part may easily distort if it is built in the horizontal plane of many RP machines. Providing support geometry may maintain the flatness during the RP build phase. Postcuring or cooling the part before the supports are removed may result in a flatter final part. Integration with automated finishing techniques may permit this to be done automatically and in a seamless manner, providing more precise, reliable results. Similar examples can be found for surface treatment of parts. This essentially treats RP as part of a complex series of process chains, contributing in different ways to the final product. If the effects of RP on specific features of a part are known within the CAD system, it should

therefore be possible to compensate for part variation or plan for processes required to meet the specification. Since RP is used for so many different applications, this represents a significant challenge to CAD researchers.

## **11.8 MEDICAL ISSUES**

This is a somewhat unusual topic in comparison with the others in this chapter. Medical models are different from engineered products in that they are extensively made up from free-form surfaces. Such surfaces are extremely difficult to describe in mathematical terms. Features derived from anatomic descriptions are not easily measured (particularly in terms of reliably identifying reference points) and boundaries are very vague. This poses two related problems as far as RP is concerned:

- Lack of regularity in models makes it very difficult to optimise the build parameters when major differences can exist between layers.
- Feature extraction is very difficult to perform and so it is difficult to predict build behaviour.

Current medical scan technology is relatively inaccurate when compared with the capability to reproduce models using RP. As the use of models becomes more popular for medical applications, and as technology like magnetic resonance imaging improves, this gap will close. RP will then be pressured to ensure accurate medical models are possible. This means that techniques for improved control, particularly responsiveness from layer to layer, must be developed. Pre-processing of RP files will permit RP systems to adjust their build strategies and control parameters to adapt to changes in surface area, wall thickness, mass, etc. This will inevitably improve the quality of all types of parts and not just medical ones. Research in this area will therefore benefit RP technology as a whole.

## **11.9 SCALE**

There are now a number of researchers focusing on unusually small or unusually large components using RP techniques. Generally, RP systems build parts in the order of a few mm to a few hundreds of mm. This is achieved using standard automation technology. Scaling upwards or downwards from this range requires greater attention to hardware and software.

Building large parts using layer-based techniques requires close attention to materials handling issues. Maintaining system stability so that vibration, mass, thermal expansion (and other factors that affect accuracy) are minimized will require significant attention. As parts become larger, the energy issues in terms of creating parts using additive approaches also become significant. Strategies for creating parts with the correct external appearance but internal structures focused on minimizing energy input must be adopted. The use of, for example, hollow structures may be desirable, but mechanical properties must be factored into their overall performance.

RP for building small parts, of less than 1 mm for example, will also require close attention to materials handling. While stability will still be a problem, the focus will be on resolution and accuracy. Again, software developers will be heavily involved in control applications. Material variations, like bubbles in resin and variable density resulting from thermally formed processes, can seriously affect the performance of such small parts. Materials modelling packages will not only be important for part design but also for predicting the ability of machine set-up parameters.

## **11.10 VIRTUAL REALITY**

Virtual reality techniques are already used extensively in CAD applications and to simulate machines and processes. Since RP technology has many parameters that can vary and thus affect the outcome of parts, it is not difficult to see how VR can contribute as a modelling tool for RP processes. The aim here is to enhance software for RP so that better parts can be produced more easily and efficiently. In this sense, VR is an ally in the development of RP technology.

VR can also be a competitor to RP technology. This is particularly true in the early stages (i.e. conceptualization stages) of product development. As VR improves in the ability to model, so RP is forced downstream into application areas that require physical parts (i.e. tooling and production). Software systems must therefore be developed to accommodate these areas. This emphasizes the importance of this book since these software packages must assist the traditional CAD models, by adding, manipulating, and managing information to enhance and simplify these downstream activities.

## **11.11 CONCLUSION**

The conclusion is the same as mentioned in the introduction to this book: there are many problems still to be solved, we are only in the early stages of identifying those problems, and this book is a result of the increased interest in RP and how to use it effectively and efficiently. There are undoubtedly problems that have not been mentioned and ones that remain undiscovered. We see here an attempt at identifying the most important ones along with possible solutions. Let's see if you, the reader, can do better.

# Index

- .STL file
  - errors 346
  - format 344
- .STL files, correction of 341
- 3DP 365
- A point cloud 286
- ACIS solid modelling kernel 179
- Active acquisition mode 290
- Active data acquisition 287
- Active server page (ASP) 42
- Adaptive
  - algorithms 131
  - slicing 86, 137, 368
- Additive material feature 237, 238
- Advanced adaptive slicing 147
  - method 143
- Aesthetic characteristics 307
- Analogue sensors 291, 292
- ASP, active server page 42
- ASTM (American Society for Testing and Materials) 67, 68
- Automatic location 290
- Backface/misoriented facet mode 351
- Basic functionality 9
- Benchmark 58, 366, 368
  - database 54
  - generalized 68, 69
  - geometric 64, 65, 69
  - mechanical 64, 66
  - parts 59
  - process 64, 74, 75
  - types of RP&M 64
- Benchmarking 3, 50, 57, 58, 368
  - process 76, 79
- Binary space partitioning (BSP) 165
- Boundary 162
  - edges 308
  - representation (B-rep) 347
- B-rep (boundary representation) 347
- BSP (Binary space partitioning) 165
- B-spline 246, 247
  - representation 246
- Build
  - platform 364
  - styles 365
- CAD/CAM 130
- CAD/CAM/RP systems 153
- CAD kernel 368
- Capital investment 96
- Cell tree 162
- CGI 287
- Chordal tolerance 345
- Chrome plated mode 352
- Class-A surfaces 305
- Client 20
- Client/server 19, 21
- Clipping plane 353
- CMM (Co-ordinate measuring machine) 76
- CMY (cyan magenta yellow) 160
- Collaborative virtual environment 24
- Colour
  - representation 158, 159
  - RP 367
- Coloured contours 354
- Colourspace 159
- Commercial digitization systems 298



- Commission Internationale de l'Eclairage (CIE) standard 159
- Complexity 216
- Composite structure 33
- Compressive strength 67
- Computer aided design/manufacture 130
- Concentricity 72
- Constructive feature operations 228
- Contact sensors 283, 287, 290
- Continuous field model 161
- Contour
  - model 135
  - support 134
- Contours
  - generation 275
  - of the part 288
- Control comparison 283
- Convex polyhedral chain 165
- Co-ordinate measuring machine (CMM) 76
- Correction of .STL files 341
- Creep 68
- Curl 67
- Curling 66
- CybergRP 22
- Cylindricity 72
- Data
  - acquisition, active 287
  - technique of 287
  - exchange 368
    - format 368
    - standard 368
  - interchange 367
  - pre-processing 317
  - systems, web-based 31
- Database 1, 3, 54, 113, 120, 366, 368
  - system 98
- Database, rapid prototyping 27
- Databases 124
- Decision support 80, 368
  - problem technique (DSP) 99
  - system (DSS) 53, 81, 367
- Decision theory 102
- Deep cavities 293
- Delaunay triangulation 300
- Design
  - feature interaction 257
  - methodology 227
  - philosophy 188
  - prototype 68
  - visualization 16
- Destructive combined mode 287
- Diffusion 246
- Digitization 283, 286, 287
  - 'commercial systems' 298
- Dimensional control 325
- Direct face neighbourhood 245
  - alteration 228, 256
- Direct manufacture 369
- Direct material deposition 181
- Direct slicing 86
- Distant and large-sized objects 293
- Distributed pre-processor based on network 18
- DSP (Decision support problem technique) 99
- DSS (Decision support system) 53, 367
- Edge-based growing approach 308
- Edit mode 357
- Effect
  - laddering 16
  - staircase 16, 90
- Electromagnetic waves 290
- Emission of light 290
- Error
  - map 325
  - percentage volumetric 139, 141
  - based triangulation 301
- Errors
  - mode 352
  - stair-stepping 130, 137
- Exchanging data 368

- Expert system shell 113
- Fabrication 75
  - feature-based 256
  - methodology 227
  - parameters 89
- Facet functions 350
- FDM (Fused deposition modelling)
  - 107, 112, 363, 364, 365
- Feature interaction volume (FIV)
  - 251, 256
- Feature recognition 227, 257
- Feature-based
  - design 227, 256
  - and fabrication
  - methodologies 227
  - fabrication 256
  - slicing 250
- Features 227, 232
  - geometric 65, 70
- FGM (functionally gradient materials) 183, 225
- Fillets 305
- Finishing method table 40
- FIV (Feature interaction volume)
  - 251, 256
- Flatness 72
- Flying mode 356
- Flywheel 204
- Form feature 232, 233, 235, 250
- Functional
  - characteristics 307
  - prototype 68, 125
  - prototypes 106
- Functionalities 14
- Functionality 10, 15
  - basic 9
  - pre-process 7
- Functionally gradient materials (FGM) 183, 225
- Functions
  - facet 350
  - mode 350
- Fused deposition modelling (FDM)
  - 107, 112, 363, 364, 365
- Gamut 159
- Generalized benchmark 68, 69
- Geometric
  - benchmark 64, 65, 69
  - blending material 199
  - boundary evaluation 241
  - curvature effect 231, 250
  - features 65, 70
  - prototype 68
- Geometry curvature effect 231
- Geosciences modelling 161
- Global visibility 327
- Grab mode 357
- Half-spaces 165
- Head mounted display (HMD) 356
- Heterogeneous
  - object 225
    - fabrication 274
    - modelling 227
    - realization 227
  - solid modelling 264
- Hierarchical
  - space partitioning 165
  - structure 33
- HMD (head mounted display) 356
- Homogenization 189
  - design method 230
- HTML 42
- Hypervolume interpolation 177
- Implicit
  - methods 168
  - representation 172
- Improved adaptive slicing methods 142
- Inner loop 133, 139
- Inspection of complex parts 285
- Integration 80
- Interface, server-client 21
- Internal
  - geometry 287
  - part 293

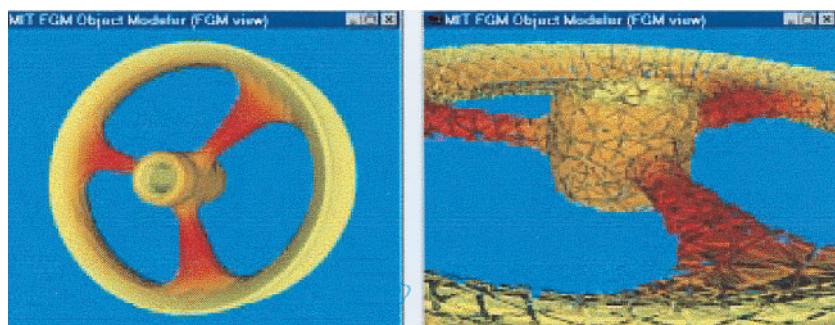
- Internet 3, 18, 24, 365
- Interpolation 168
- IVECS, the Interactive Virtual Environment for the Correction of .STL 341
- JAVA 21
- Kernels 368
- Key considerations 75
- Laddering effect 16
- Laser 296, 308, 324
  - plan 297
  - plane 302
  - point 297
- Layer-based fabrication processes 2
- Layered manufacturing (LM) 7, 8, 227, 230, 263
  - data (LMData) 228
- Levels of grey 288
- Linear structure 32
- Light, emission of 290
- LM (Layered manufacturing) 7, 8, 227, 230, 263
- LMData (Layered manufacturing data) 228
- Local visibility 326
- Loop
  - inner 133, 139
  - outer 133, 139
- Management of RP technology 3
- Manufacturability analysis 79
- Map of the error 285
- Material 92, 235
  - discontinuities 203
  - feature 234
  - features 250
    - partition 237, 238
    - gradation effect 232, 250
  - process
    - chain selection (MPCS) 97
    - selection (MPS) 97
    - properties 366
    - region forming 241
    - selection 80
    - shrinkage 90
    - space 191
- Measurement 75, 283
- Mechanical
  - benchmark 64, 66
  - properties 66
- Medical
  - imagery 292, 293
  - imaging 163
  - models 370
- Merging 308
- Microsoft Access 43
- MMA-REP 188
- Mode functions 350
- Mode,
  - backface/misoriented facet 351
  - chrome plated 352
  - edit 357
  - errors 352
  - flying 356
  - grab 357
  - transparent 351
  - visualization 350
- Model
  - contour 135
  - data interchange 2
  - visually simulated 16
- Modelling 158
  - cloud data 298
  - fused deposition 107
  - multi-jet 107
  - multiple-material 4
- Moiré effect 294, 296
- MPCS (Material process chain selection) 97
- MPS (Material process selection) 97
- Multi-jet modelling 107
- Multi-material 155
  - modelling 4
- Multiple
  - material systems 369

- range images 300
- views 308
- Network 18, 24
- Network-based pre-processor 15
- Networks of fringes 295, 296
- Non-contact sensors 283, 287, 292
- Non-conventional manufacturing 130
- Non-linear structure 33
- Non-metal objects 293
- Non-textured materials 290
- Nuclear magnetic resonance 293, 294
- Octrees 162
- Odd winding rule 133
- Off-line programming module 324
- Open issues 4
- OpenInventor 349
- Optimization
  - of RP processes 4
  - problem 205
- Oracle 43
- Orientation 85
- Outer loop 133, 139
- Outside contour 145
- Packaging of parts 364
- Packing problem 364
- Parameters
  - fabrication 89
  - process 89
- Parametric representations 298
- Part orientation 123
- Partition material feature 237, 238
- Passive
  - acquisition mode 288
  - data acquisition 287
- Patching of a hole 355
- Path planning 87
- Percentage volumetric error 139, 141
- Photogrammetric chambers 289
- Photogrammetry 289
- Physical contact 290
- Physics based B-spline
  - heterogeneous object modelling 256
- Physics (diffusion)-based modelling 245
- Physics-based modelling 228
- Planning, process 77, 78
- Point-based
  - curve model 319
  - segmentation 301, 315
- Pointer 133
- Point-to-point sensors 291
- Polyhedral
  - mesh 298
  - meshes 300
- Post-processing 88, 123
- Pre-process functionality 7
- Pre-processing 11, 13, 88
- Pre-processor 7, 8, 15, 23
  - network-based 15
- Process
  - benchmark 64, 74, 75
  - benchmarking 76, 79
  - chains 75, 85
  - parameters 89
  - plan 96, 98, 112
  - planning 77, 78, 79
- Properties, mechanical 66
- Prosthesis 225, 248
- Prototype 68
  - design 68
  - functional 68, 106, 125
  - geometric 68
  - technical 68
- Quadtree 162
- Quotation support 96
- R/m sets 179
- Radar 292
- Rapid prototyping (RP) 8, 129
  - and manufacturing (RP&M) 57
  - applications table 36
  - company 52
  - database 27

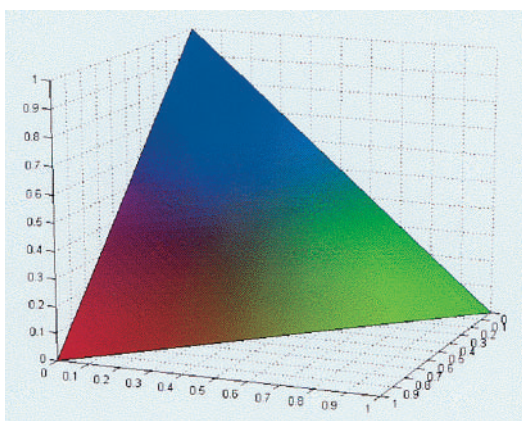
- processes 77
- data collections 41
- decision support 50
- fabrication
  - cost table 40
  - time table 39
- machine 45, 46, 49, 51, 52
  - table 35
  - type table 37
- material type table 40
- materials table 36
- mechanical properties table 38
- operating systems 3
- processes 77
  - optimization of 4
- surface qualities table 38
- system 43
- technology, management of 3
- tolerance table 39
- vendor 47, 49
- vendors 48
  - table 37
- website design 32
- Ray tracing 163
- Real visibility 328
- Refined feature volume (RFV) 251
- Repeatability 72
- Representation 158
  - schemes 227
- Representation, colour 158, 159
- Reverse engineering (RE) 4, 284, 287, 298
- R-functions 174
- RGB 160
- Roundness 72
- RP (Rapid prototyping) 8, 129
- RP&M (Rapid prototyping and manufacturing) 57
- RPDSS (Rapid prototyping decision support system) 27, 29, 50
- SBs (Service bureaus) 121
- Scanning strategy 325
- SDK (Software development kit) 113
- Segment-and-fit 298
- Selection decision support problem (sDSP) 99
- Selective laser sintering 112, 363
- Sensor
  - manufacturers 287
  - technologies 285
- Sensors, analogue 291, 292
- Server 20
- Server–client interface 21
- Service bureaus (SBs) 121
- Shading 349
- Shepard's method 169
- Shrinkage 66, 67
- Simulation 284
  - and process planning 4
- SLA (Stereolithography) 112, 363, 365
- Slicing 85, 274
  - adaptive 86, 137, 368
  - advanced 147
  - improved 142
  - algorithms 152
  - direct 86
  - feature-based 250
  - process 132, 136
- SLS (Selective laser sintering) 112, 363
- Smoothness 72
- Software development kit (SDK) 113
- Sorting process 133
- Spatial filtering method 309
- Spherical geometry 326
- SQL server 43
- Staircase
  - cuspl 17
  - effect 16, 90, 230
- Stair-stepping errors 130, 137
- Standardized measurement 76
- Standards 58, 367, 368
  - (benchmarking) 367
  - authority 367
- State-of-the-art reverse engineering 287

- STEP/ISO 10303 181
- Stereolithography (SLA) 107, 112, 363, 365
- Stereometric chambers 289
- STL 8, 9, 12, 19, 20, 23, 77, 367, 368
  - file 367
  - format 131
  - model 134, 135
- Storage cost 216
- Straightness 72
- Strength
  - compressive 67
  - tensile 67
- Structure
  - composite 33
  - hierarchical 33
  - linear 32
  - non-linear 33
- Structured light 294
- Subtractive material feature 237, 238
- Support
  - contour 134, 135
  - structure 85
  - structures 87, 365
- Supported/supportless building 86
- Surface
  - finish 75, 364
  - geometry 149
  - normal 142
    - vectors 145
  - rebuilding 303
  - reconstruction 284, 298, 301
  - roughness 16, 17, 90
  - slopes 149
- Symmetry 72, 305
- Synthesized features 237
- System, database 98
- Systems, web-based 99
- Technical prototype 68
- Technique of data acquisition 287
- Telemeters 293
- Tensile strength 67
- Tensor product B-spline volume 246
- Tessellation 346
  - representation 131
- Theory
  - decision 102
  - utility 100, 102, 103, 105
- Thermal distortion 363
- Three-dimensional mouse 356
- Time of flight 292
- Tolerance, chordal 345
- Tomography 293
- Topological information 131
- Touch probe 290
- Transfinite interpolation 173
- Transition zone 215
- Transparent mode 351
- Triangular
  - facet 133
  - mesh 310
- Triangulation 296
  - process 308
- Two-sided face neighbourhood 241
- Types of RP&M benchmarks 64
- U21 1
  - universities 1
- Ultrasound 293
- Underwater area 293
- Unifying layered manufacturing data 181
- Universitas 21 1
- User interface 350, 365
- Utility theory 100, 102, 103, 105
- Utility theory-based selection
  - decision support problem 100
- Verification 323
- Virtual
  - prototyping (VP) 4, 284, 341
  - reality 355, 371
- Visibility 325
  - calculation algorithm 324
- Visualization 15, 16

design	16	bins	309
mode	350	VR	371
simulated model	16		
Volume		Warpage	91
features	237	Water-tight	345
rendering	163	Weakness/difficulties	90
Volumetric	155	Web-based	
representations (geometry)	162	data systems	31
Voroni		design tools	42
diagram	194	system	99, 116
polygon	194	Website design, RP	32
Voxel	162		

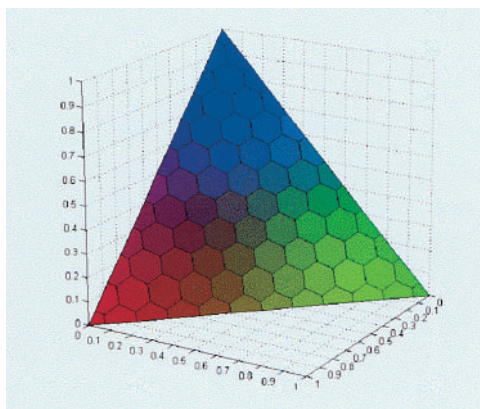


**Fig 6.11** Decomposition of a solid in cells (51)

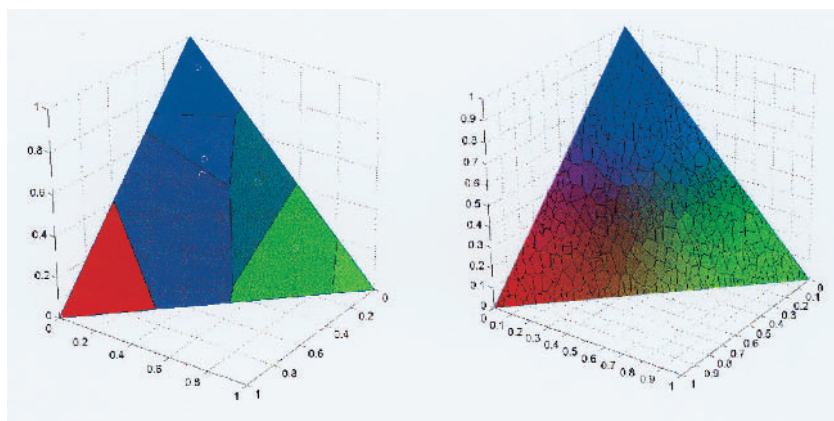


**Fig 6.13** The continuous material space  $V$  for three materials

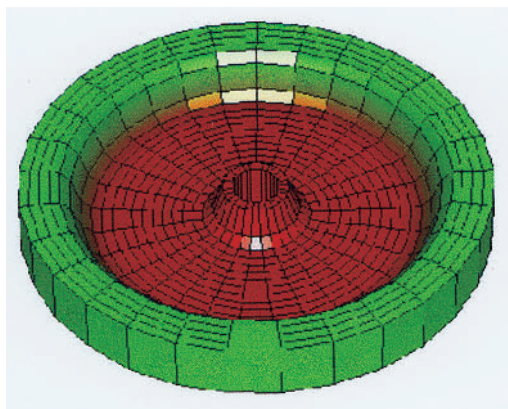




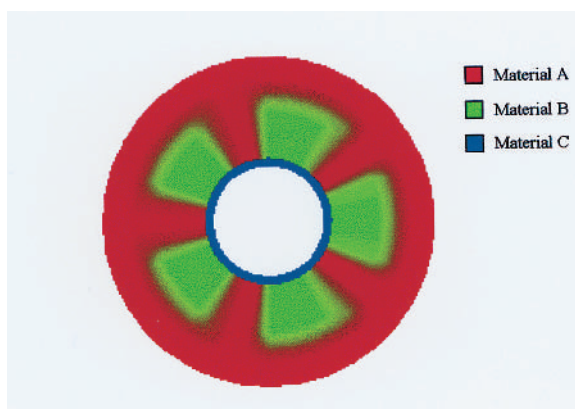
**Fig 6.14** The discretized material space  $V$  for three materials



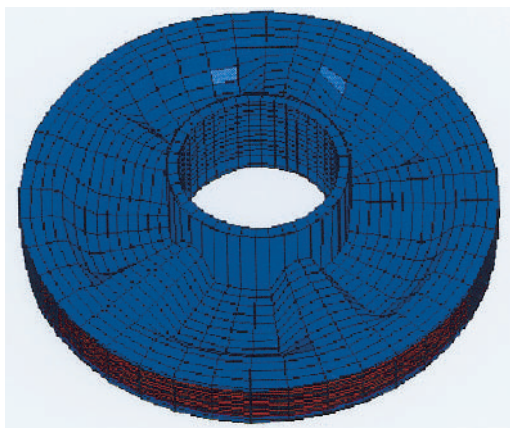
**Fig 6.15** Two non-uniformly discretized parameter spaces  $V$



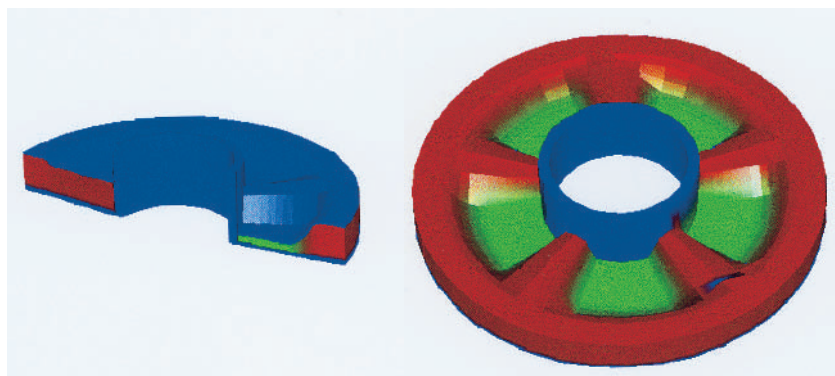
**Fig 6.29** Three-dimensional view of the heterogeneous flywheel



**Fig 6.30** Heterogeneous flywheel composed of three materials



**Fig 6.37 Three-dimensional heterogeneous flywheel**



**Fig 6.38 Cross-sections and details of three-dimensional flywheel**