A Second Generation Computer Aided Design System for Prosthetics and Orthotics

by

Rowland Paul Travis

A Thesis Submitted for the Degree of Doctor of Philosophy in the Faculty of Engineering

UNIVERSITY OF LONDON 1991

Department of Mechanical Engineering University College London Torrington Place LONDON WC1E 7JE
Dedicated to my parents and to Cathy
Abstract

This thesis begins with an introduction to Computer Aided Design (CAD) and its common uses, particularly in engineering and bioengineering. A computer based system is documented which was developed at UCL to design above-knee prosthetic sockets. The first objective in the system's development was to provide an automated facility capable of taking surface measurements of a residual limb and manipulating these data to produce a socket shape using conventional design philosophy.

The UCL system is regarded as a First Generation system in that it is restricted in its possible applications and it has no mathematical understanding of the surface it designs, thereby making large scale manipulations of the surface cumbersome and difficult to quantify. This thesis seeks to develop a Second Generation system, generally applicable in prosthetics and orthotics, including a complete description of the surface designed and enabling straight-forward manipulation of the surface.

Applications of a CAD system in prosthetics and orthotics involve modelling an existing shape, and so a review of data capture techniques and a developed software tool for examination of the data captured are presented. After a review of surface modelling techniques which fails to yield a suitable method, a surface model for general application is developed together with a method for reducing the data captured to the information necessary for the model. The ability of the model to represent an existing shape is demonstrated with appropriate examples. The developed data examination tool, surface model, data reduction method and three-dimensional graphics software form the Second Generation system.

Manipulation of an original shape according to known rules is a procedure frequently followed in prosthetics and orthotics, and since a CAD system is often used to mimic a conventional design philosophy, the Second Generation system has been developed with the ease of manipulation paramount. The suitability of the system for manipulation is demonstrated by application to an orthotic project where the rules have been quantified in terms appropriate to the system.
Acknowledgements

I would like to thank all those who contributed suggestions to the current work. The advice of Ms.M.Lord, Dr.R.Collins and Dr.M.Dewar has been particularly helpful.

I would also like to thank my wife and friends without whose support and encouragement this work would never have been completed.
## Table of Contents

Table of Figures ................................................................................................ 15

Table of Tables ................................................................................................ 25

Chapter 1  
**Introduction**  ............................................................................................ 27  
1.1 Motivation ......................................................................................... 28  
1.2 Contributions of the current study ........................................... 36  
1.3 Structure of the current thesis .................................................. 37

Chapter 2  
**A First Generation CAD System - The UCL Computer Aided Socket Design System**  .................................................................................................................. 39  
2.1 Introduction ............................................................................... 40  
2.2 Conventional above-knee socket design ..................................... 41  
2.3 Philosophy of the system .......................................................... 47  
2.4 Brim shapes and sizes .............................................................. 49  
2.5 Capture and storage of residual limb shape ......................... 54  
2.6 Blending brim and residual limb shapes ............................ 58  
2.7 Shape visualisation ................................................................. 61  
2.8 Modification of the socket shape ........................................... 65  
2.9 Shape carving and socket production .................................. 69  
2.10 The user interface ................................................................. 70  
2.11 Remarks ................................................................................... 71

Chapter 3  
**Data Capture Techniques** .................................................................... 73  
3.1 Introduction ............................................................................... 74  
3.2 Tactile methods ......................................................................... 75  
3.3 Optical methods .......................................................................... 79  
3.4 Ultrasonic and other radiation methods .................................. 89  
3.5 Remarks ................................................................................... 91

Chapter 4  
**VIEW3D - An Editor for Three-Dimensional Data Files** ................... 95  
4.1 Introduction ............................................................................... 96  
4.2 Data format ............................................................................... 97
4.3 Three-dimensional viewing ........................................ 98
  4.3.1 Three screen viewing ........................................ 98
  4.3.2 Rotations .......................................................... 99
  4.3.3 Viewing cross sections and Visibilities .................. 100
  4.3.4 Cross sections .................................................. 102

4.4 Three-dimensional editing ........................................... 103
  4.4.1 A three-dimensional cursor ..................................... 103
  4.4.2 Editing individual points ....................................... 104
  4.4.3 Editing using the Visibilities .................................. 105

4.5 Remarks ........................................................................ 106

Chapter 5 A Review of Computer Aided Geometric Design and
Surface Modelling Techniques ........................................ 107
  5.1 Introduction ............................................................. 109
  5.2 Curves in two and three dimensions ......................... 111
    5.2.1 Definition of a curve ........................................... 112
    5.2.2 Standard polynomial form .................................... 115
    5.2.3 Bezier curves .................................................... 116
    5.2.4 Hermite curves .................................................. 126
    5.2.5 B-spline curves .................................................. 127
    5.2.6 Matrix representation of a curve ......................... 136

  5.3 Interpolation with curves .......................................... 138
    5.3.1 The interpolation requirements .............................. 138
    5.3.2 Aitken's algorithm ............................................. 141
    5.3.3 Hermite curve interpolation .................................. 144
    5.3.4 Bezier curve interpolation .................................... 145
    5.3.5 B-spline curve interpolation .................................. 147
    5.3.6 Continuity ....................................................... 148

  5.4 Surfaces in three dimensions ...................................... 149
    5.4.1 Definition of a surface .......................................... 150
    5.4.2 Coons patches ................................................... 153
    5.4.3 Tensor product surfaces ....................................... 156
    5.4.4 Tensor product Bezier patches ............................... 157
    5.4.5 Tensor product Hermite patches ............................ 164
    5.4.6 Tensor product B-spline surfaces ........................... 166
6.4.3 The tangent plane at a vertex of order five ........ 222
6.4.4 Tangent vectors at a vertex of order four ............ 224
6.4.5 Tangent vectors at vertices of orders three and five 230
6.4.6 Tangent estimation from intermediate information ............................................................... 233

6.5 Twist estimation .................................................................................. 234
6.5.1 Twist vectors at a vertex of order four .................. 235
6.5.2 Twist vectors at vertices of order three and five 236
6.5.3 Twist estimation from intermediate information 238

6.6 Formulae for the Bezier points of the new surface model 238
6.7 Algorithms for modifications of the new surface model 243
6.8 Remarks ................................................................................... 245

Chapter 7
Adaption of the New Surface Model to Particular Situations and the Fitting of Data to it ................................. 247
7.1 Introduction ............................................................................. 248
7.2 Adaption of the new surface model to particular situations .................................................................................. 248
7.2.1 The prosthetic socket ................................................ 253
7.2.2 The whole foot ........................................................... 254
7.2.3 The orthotic insole ........................................................... 255
7.3 Marker points .......................................................................... 256
7.4 Fitting the data to the new surface model .................... 257
7.4.1 Structured data files .................................................. 257
7.4.2 Dividing up the data and averaging ..................... 258
7.4.3 Fitting the data by curve fitting techniques .......... 262
7.4 Remarks ................................................................................... 267

Chapter 8
Experimental Results from Application of the New Surface Model ................................................................. 269
8.1 Introduction ............................................................................. 270
8.2 Evaluation of the point-model distance ................. 270
8.3 The UCL CASD system and the new surface model ...... 273
8.4 A whole foot model using data averaging techniques ... 278
8.5 An orthotic insole model using curve fitting techniques 281
8.6 Modifications applied to the orthotic insole model ...... 283
8.7 Remarks ................................................................................... 284

Chapter 9  MODEL - The Surface Display Software ............... 287
9.1 Introduction ............................................................................. 288
9.2 Format of the data file ............................................................ 288
9.3 Surface display modes ........................................................... 290
  9.3.1 Wire frame displays ................................................. 290
  9.3.2 Shaded solid views .................................................. 292
9.4 Remarks ................................................................................... 295

Chapter 10  Conclusions ............................................................... 297
10.1 Conclusions from the current work ................................. 298
10.2 Suggestions for future work ................................................. 302

List of the Author’s Publications .............................................. 305

References .................................................................................. 307
Table of Figures

Chapter 1 Introduction

Figure 1.1 The constituent parts of an integrated CAD/CAM system for prosthetics and orthotics. ......................................................... 33

Chapter 2 A First Generation CAD System - The UCL Computer Aided Socket Design System

Figure 2.1 A Hosmer brim which is adjustable at its medial and lateral extremes. ................................................................. 41
Figure 2.2 An unadjustable Blatchford 'European' brim................. 42
Figure 2.3 Measuring the antero-posterior (AP) dimension. ....... 43
Figure 2.4 Measuring the medio-lateral (ML) dimension. .......... 44
Figure 2.5 Measuring the circumference (Circ) dimension....... 44
Figure 2.6 The correspondence between the dimensions and the brim shape. ............................................................................. 45
Figure 2.7 The anatomical relationship to the brim shape. ........... 50
Figure 2.8 Initial set up of a Standard brim shape indicating the Medial and Lateral openings, M and L. ................................ 52
Figure 2.9 The posterior-medial portion of a brim used in casting. ......................................................................................... 55
Figure 2.10 The prosthetist marks the centre of the posterior aspect as a reference point. ......................................................... 56
Figure 2.11 A measurement grid for residual limb illustrating the terms 'slice' and 'strip'. .......................................................... 57
Figure 2.12 The measurement system. ........................................ 58
Figure 2.13 The profile of one strip at the join between brim and measurement (a) before and (b) after blending. ............... 60
Figure 2.14 The effects of different smoothing masks (a) \( \left( \begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array} \right) \), 
(b) \( \left( \begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array} \right) \), (c) \( \left( \begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array} \right) \) ........................................................................ 61
Figure 2.15 Four adjacent data points define a facet. ................. 62
Figure 2.16 In 'High Resolution' mode, original data points (a) are interpolated first in slices (b) and then in strips (c) to yield four facets for each original facet. .............................. 63
Figure 2.17 The interpolation algorithm can cause an error near a large discontinuity (a), but this is readily corrected (b).

Figure 2.18 The planes for Flexion/Extension and Adduction/Abduction shown in the slice containing the reference point. The angle, $\theta$, of a point or strip relative to the reference point is also indicated.

Figure 2.19 The effect of Flexion/Extension and Adduction/Abduction on one strip: (a) the original data; (b) the five slices below the shelf are expanded; (c) each point on and above the shelf is moved and the slices below the shelf are stretched as necessary; (d) interpolation yields the final data on the same horizontal planes as the original data.

Figure 2.20 A trial fitting of a patient with a socket designed by the CASD system.

Chapter 3 Data Capture Techniques

Figure 3.1 A tree showing classes of data capture methods.

Figure 3.2 Typical data sets obtained from a hand-held sensor using (a) pre-determined structure and (b) free-streaming modes.

Figure 3.3 The patterns from two gratings at an angle to one another form Moiré fringes.

Figure 3.4 In Photogrammetry, the distance $b$, from the camera to the object, and the distance $d$ between the two camera positions should be related by $0.3 < b/d < 1.0$.

Figure 3.5 A data file from the Isis scanner, which uses a triangulation and structured light method.

Figure 3.6 A silhouetting method for capturing data from a leg requires a lens to ensure that incoming horizontal rays are focussed at the camera.

Figure 3.7 A typical image from a silhouetting method for a leg.

Figure 3.8 In a silhouetting method, tangent lines define a convex area in which the object lies.
Figure 3.9 A cross-section which cannot be picked up by a silhouetting method. ......................................................... 89

Figure 3.10 A silhouetting technique can be used to calculate radial values for data at regular angular intervals. ........... 90

Chapter 4 VIEW3D - An Editor for Three-Dimensional Data Files
Figure 4.1 The manner in which VIEW3D displays a data file comprised of points on the edges of a bevelled cube. Note that in this instance data points on the back of the cube are obscured by points on the front. ................................. 99

Figure 4.2 The bevelled cube data file of the previous figure is rotated about an axis perpendicular to the top left quadrant. The previously obscured data points are now visible. ................................................................. 100

Figure 4.3 Visibilities are defined as the maximum and minimum extents of the data file which are currently in view. .................................................................................................................. 101

Figure 4.4 Alteration of the Visibilities affects the portion of the data file currently in view. Each Visibility appears in two screens and affects the view in the third. .............................. 102

Figure 4.5 Examples of cross-sections of a data file from a foot. 103

Figure 4.6 Movement of the three-dimensional cursor in one quadrant affects its position in one direction only in each of the other quadrants. Here the movement is in the upper left quadrant. ................................................................. 104

Figure 4.7 The Visibilities can be used to delete all points which are not currently in view in all three quadrants. ........... 105

Chapter 5 A Review of Computer Aided Geometric Design and Surface Modelling Techniques
Figure 5.1 The Bernstein polynomials of degree $n = 4$ .......... 117

Figure 5.2 Examples of Bezier curves with their control polygons. ................................................................. 118

Figure 5.3 A Bezier curve lies within the convex hull of its control points. ................................................................. 119
Figure 5.4 The de Casteljau algorithm for \( n = 4 \). .......................... 120

Figure 5.5 Degree elevation of a cubic Bezier curve to a quartic Bezier curve. The curve itself is unaltered. ......................... 123

Figure 5.6 In subdivision of a Bezier curve, the new control points \( c_i \) are obtained from the de Casteljau algorithm for the subdivision point. ........................................... 124

Figure 5.7 A cubic Hermite curve. ........................................ 126

Figure 5.8 Examples of B-spline curves with their control points indicated, and demonstrating that the curves can be open or closed. ................................................................. 129

Figure 5.9 Each portion of a B-spline curve lies within the convex hull of the relevant control points. .................................... 130

Figure 5.10 Local control of a B-spline curve means that if one control point is repositioned only the 'nearby' curve portions are affected. ................................................................. 131

Figure 5.11 The de Boor algorithm for calculation of a point on a B-spline curve. ................................................................. 132

Figure 5.12 The knot insertion algorithm for a B-spline curve uses the points generated by the first step of the de Boor algorithm. ................................................................................ 133

Figure 5.13 The Bezier points for a B-spline curve. ....................... 134

Figure 5.14 A common estimate for tangent direction at point \( p_i \) is parallel to the line \( p_{i-1}p_{i+1} \) .................................................. 139

Figure 5.15 An interpolating smooth curve cannot possess the variation diminishing or convex hull property since there must be a point \( p_i \) and a plane through that point which has all other points on one side of it, and cuts the curve more often than it cuts the defining polygon. ...... 142.

Figure 5.16 The 'High Resolution' choice in the UCL CASD system uses Aitken's algorithm to determine intermediate values of radius. Two calculations are shown. ................................................................. 144

Figure 5.17 For two Hermite cubic curves to join smoothly they must have a common end point and parallel tangents at that point. ................................................................. 145
Figure 5.18 For two Bezier curves to join smoothly requires a common end point and three collinear Bezier points. ... 146

Figure 5.19 B-spline curves can interpolate to points \( p_i \), after control points \( d_i \) have been determined. ........................................ 148

Figure 5.20 A Coons patch is a surface which interpolates its four boundary curves. ......................................................... 153

Figure 5.21 A tensor product Bezier patch with its control points and control polygon. .......................................................... 158

Figure 5.22 The deviation of the quadrilateral of control points near a corner of a Bezier patch is proportional to the twist at that corner. .......................................................... 162

Figure 5.23 A Hermite cubic tensor product patch. .................... 164

Figure 5.24 A Bezier triangle with its control points. .............. 168

Figure 5.25 If the rows of control points across the join of two adjacent Bezier patches are collinear, and in the same ratio, then the join is tangent continuous. ......................... 176

Figure 5.26 Tensor product B-splines can interpolate points \( p_{ij} \), but this requires determination of the control points \( d_{ij} \). .......................................................... 177

Figure 5.27 The three possible rectangular situations - sheet, cylinder and torus. ................................................................. 180

Figure 5.28 A foot shape cannot be modelled over a plane domain. A non-convex domain is required. ....................... 181

Figure 5.29 A regular triangular mesh for Farin's method. ...... 183

Figure 5.30 An arbitrary triangular mesh which requires Piper's method. .......................................................... 183

Figure 5.31 Examples of faces with other than four sides in largely rectangular meshes. ..................................................... 184

Figure 5.32 Examples of vertices of orders other than four in largely rectangular meshes. ............................................... 186

Figure 5.33 An example of a large scale manipulation. Here two rows of interpolation points are moved. .................... 188

Figure 5.34 An example of a local manipulation for a tensor product Bezier patch surface. Here one interpolation point is moved but the shaded area of the surface is affected. ........................................................................ 189
Chapter 6  A New Surface Model for General Application in Prosthetics and Orthotics

Figure 6.1 Example meshes for parts of the human anatomy. Vertices of orders other than 4 are indicated. ................................................................. 199

Figure 6.2 Example meshes for closed shapes - a sphere and an ellipsoid. ........................................................................................................ 199

Figure 6.3 A possible spiralling boundary between two adjacent rows of patches in a non-rectangular situation. ............................ 200

Figure 6.4 The notation for parameters and Bezier points about a vertex $p$. ........................................................................................................ 203

Figure 6.5 The Bezier points for a system of quartic patches around a vertex of order 5. The points undetermined by position and tangent continuity conditions with other patches are indicated by crosses. ..................................................... 204

Figure 6.6 A geometrical interpretation of (6.2) is that the indicated vectors are equal. ................................................................. 205

Figure 6.7 If the vectors $q_i$ are of equal magnitude and at equal angles, then $q_i = \lambda (q_{i-1} + q_{i+1})$ is anticipated for some $\lambda$. ................................................................. 209

Figure 6.8 The effect of modifying the Bezier points about a vertex of order 3 to satisfy the tangent conditions at the vertex. ........................................................................................................ 210

Figure 6.9 The effect of modifying the Bezier points about a vertex of order 5 to satisfy the tangent conditions at the vertex. ........................................................................................................ 211

Figure 6.10 Examples of the position of the twist Bezier points $b_{ij}^{(k)}$ about vertices of orders 3 and 5. ................................................................. 217
Figure 6.11 The relabelling of the points required for tangent estimation at a point on the edge of the mesh. .......................... 223

Figure 6.12 The projection of the tangent $t$ at $p_0$ in the direction of $p_3$ is the fraction $\lambda$ of the vector $p_3 - p_0$. .......................... 225

Figure 6.13 The projection of the vector $b_1 - p_0$ onto $q_3$ should be less than $\frac{1}{2} q_3$ and more than $rq_3$. .......................... 227

Figure 6.14 The direction of the tangent $t$ at $p_0$ is in the direction of the projection of $p_1$ into the tangent plane at $p_0$. ........ 231

Figure 6.15 The positions on the mesh of intermediate points which may be known, and which can be used for tangent and twist estimation. Here $m = 3$ in the notation of this section. .................................................................................. 234

Figure 6.16 The notation for estimating the twist at a vertex of order four. ................................................................. 235

Figure 6.17 The notation for estimating the twist at a vertex on the edge of a mesh. .......................................................... 236

Figure 6.18 The relationship between tangent and twist vectors and Bezier points for establishing explicit formulae for the points. .................................................................................. 239

Chapter 7  Adaption of the New Surface Model to Particular Situations and the Fitting of Data to it

Figure 7.1 A rectangular mesh does not require sharp right-angled corners because the shape of patches near the corners can be adjusted. .......................................................... 249

Figure 7.2 A vertex of order 6 can be replaced by two vertices of order 5. Similar procedures can be adopted for vertices of higher order. .......................................................... 250

Figure 7.3 Two adjacent exceptional vertices can be separated by the introduction of a line of vertices between them. 251

Figure 7.4 A cap can be added to the end of a cylindrical mesh to give a mesh suitable for a prosthetic socket including its distal end. The exceptional vertices are indicated. .... 252
Figure 7.5 A first possible mesh for a whole foot, corresponding to 'sock' shape. The rectangular regions and exceptional vertices are indicated. ............................................................ 253

Figure 7.6 A second possible mesh for a whole foot, corresponding to 'sole' and 'upper' shape. The rectangular regions and exceptional vertices are indicated. ................................................................................. 254

Figure 7.7 A mesh for the orthotic insole with rectangular regions, exceptional vertices and size parameters indicated. .................................................................................. 255

Figure 7.8 An example mesh for a below-knee prosthetic socket constructed from data for the UCL CASD system. .......... 258

Figure 7.9 The marker points added to a whole foot data file to establish a mesh of the form of Figure 7.5. ...................... 259

Figure 7.10 The slicing procedure to establish a whole foot model of the form of Figure 7.5. ........................................ 261

Figure 7.11 A mesh for a whole foot constructed using the dividing up the data and averaging technique. .................. 262

Figure 7.12 The marker points added to an orthotic insole data file to establish a model of the form of Figure 7.7, and the subsequent orientation and slicing. ................................. 262

Figure 7.13 An example midfoot slice for curve fitting from an orthotic insole data file. ................................................. 265

Figure 7.14 The curve fitting technique for the slice of Figure 7.13. .................................................................................. 266

Figure 7.15 A mesh for an orthotic insole constructed using curve fitting techniques. ...................................................... 267

Chapter 8 Experimental Results from Application of the New Surface Model

Figure 8.1 The first stages of the iterative procedure to find the nearest point of a Bezier patch $b(u, v)$ to point $p$. ............ 271
Figure 8.2 A below-knee prosthetic socket represented by the new surface model using data from every third slice and strip of a UCL CASD data file with intermediate information used in the estimation of tangents and twists. ................................................................. 276

Figure 8.3 A below-knee prosthetic socket represented by the new surface model using data from every 4th slice and strip of a UCL CASD data file with no intermediate data used. ................................................................. 277

Figure 8.4 A whole foot model fitted by the dividing up and averaging method. The toe region is clearly unsatisfactory. ................................................................................ 280

Figure 8.5 A slice of data from an orthotic insole data file indicating one point which was not well represented by the model. ................................................................. 281

Figure 8.6 An orthotic insole model fitted by curve fitting techniques. ................................................................................ 284

Figure 8.7 The orthotic insole model of Figure 8.6 with preliminary implementation of the plantar surface eversion quantified by Foulston et al. ................................. 285

Chapter 9  MODEL - The Surface Display Software

Figure 9.1 A prosthetic socket model displayed in wire frame mode with all lines visible. ................................................................. 290

Figure 9.2 A prosthetic socket model displayed in wire frame mode with hidden lines removed. ................................................................. 291

Figure 9.3 The normal $n$ to a sub-patch is the cross product of its two diagonals $r_2 - r_4$ and $r_1 - r_3$. The intensity with which the sub-patch is shaded is determined by the cosine of the angle between its normal and the vector $l$ to the light source. ................................................................. 292

Figure 9.4 A prosthetic socket model displayed in a shaded solid view with $density = 4$. ................................................................. 294

Figure 9.5 A prosthetic socket model displayed in a shaded solid view with $density = 12$. ................................................................. 295
Chapter 10  Conclusions

Figure 10.1 The constituent parts of an integrated CAD/CAM system for prosthetics and orthotics as in Figure 1.1, but with the relevance of the work in the chapters of this thesis indicated.
## Table of Tables

### Chapter 2  A First Generation CAD System - The UCL Computer Aided Socket Design System

**Table 2.1** Comparisons between AP, ML and Circ dimensions in millimetres taken from a patient’s limb, and the corresponding measurements on a brim adjusted to fit comfortably, measured according to Figure 2.6. .......... 51

### Chapter 4  VIEW3D - An Editor for Three-Dimensional Data Files

**Table 4.1** The Format for a VIEW3D Data File ......................... 97

### Chapter 8  Experimental Results from Application of the New Surface Model

**Table 8.1** The Point-Model Distances for a typical below-knee measurement file from the UCL CASD system when fitted to the new surface model. All distances are truncated down to the nearest mm. .......................... 272

**Table 8.2** The Point-Model Distances for a typical below-knee socket file from the UCL CASD system when fitted to the new surface model. ......................................................... 273

**Table 8.3** The mean, standard deviation and maximum point-model distances of Tables 8.1 and 8.2 .................. 275

**Table 8.4** The Point-Model Distances for a whole foot model fitted to the new surface model by data averaging (a) without modifications and (b) with the heel and toe region vertices modified graphically. The results from the heel and toe regions are included in the table. ....... 279

**Table 8.5** The mean, standard deviation and maximum point-model distances of Table 8.4. .......................... 279

**Table 8.6** The Point-Model distances for an orthotic insole file fitted by curve fitting techniques. .......................... 282

**Table 8.7** The mean, standard deviation and average Point-Model distances for the orthotic insole model tabulated in Table 8.6. .......................... 283
Chapter 9      MODEL - The Surface Display Software
Table 9.1 The format of one Bezier patch contained in a data file for program MODEL. ............................................................ 289
Chapter 1

Introduction

1.1 Motivation

1.2 Contributions of the Current Study

1.3 Structure of the Current Thesis
1.1 Motivation

Since the late 1950's, when computers first had a limited ability to display graphical images, there has been a steady and consistent growth in their functionality and adaptability. Computer graphics are now used in many diverse fields, such as Cartography, Chart and Graph Plotting for Business Planning, Animation, Simulation, Process Control and Planning, Office Automation, Desk-Top Publishing and Creative Art. Many of these applications rely on the old adage that 'a picture is worth a thousand words', and computers can be good at translating large quantities of numerical or quantifiable facts into pictorial information. Because of this, it was soon appreciated that computers could be particularly useful as an aid to the process of design, and this application has grown into a field of its own, Computer Aided Design, or CAD for short. Frequently, CAD is combined with Computer Aided Manufacture (CAM) by, say, use of a computer-controlled milling machine to produce the object designed. This gives rise to the term CAD/CAM for an integrated system.

The first implementations of CAD were concerned with the application of computers to the design and manufacture of engineering components. Because of the expense of computer graphics equipment at the time, CAD was mainly advanced by its adoption into major industries such as the aircraft and automotive sectors. Here the necessity to design intricately-shaped and aesthetically pleasing surfaces acted as a catalyst to development, inspiring work by Coons and Bezier [COONS 67, BEZIER 66, BEZIER 67], and culminating in sophisticated surface modelling techniques, and ultimately, software packages such as Bezier's Unisurf package [BEZIER 86], and others [IMSL, SI]. In this context, surface modelling refers to the use of a mathematical technique to define precisely and unambiguously the shape and other characteristics, such as smoothness, of a surface. An influential discussion of the computational methods was given by Faux and Pratt, with thorough up-to-date reviews and presentations of general engineering-inspired CAD systems by Besant and Lui and Davies et al [FAUX & PRATT 79, BESANT & LUI 86, DAVIES et al 91]. Many modern CAD systems dispense completely with the need for the drawing by hand of conceptual designs since they include the ability to plot out engineering drawings.
There are many considerations which affect the design process, and various forms of analysis can be undertaken to ensure design criteria are met. It has therefore been natural for CAD systems to be developed with analytical functionality, and now most commercial CAD packages, for example AutoCad and I-DEAS, provide such analytical ability to enhance their design features [AUTOCAD, IDEAS]. Meanwhile, the price of the computer hardware measured against its performance has continually reduced so that today there are many more applications of CAD, both in engineering and in other areas such as geology, shoe pattern design for standard lasts, the clothing industry and medical and dental applications [BARSKY & GREENBERG 80, MCCARTHY & HANDSCOMB 89, FLUTTER 83, MCCCARTNEY & HINDS 89, VERGEST et al 87, WALKER 88, BRAMWELL et al 90].

Through the use of CAD systems, the computer has become a very useful designing tool. It is however, only a tool, and will not replace the designer because its functionality is restricted, and its knowledge limited to that which is explicitly built into the software used. Besant and Lui [BESANT & LUI 86] have suggested that in a CAD environment, the computer has three main functions:

1. To serve as an extension to the memory of the designer;
2. To enhance the analytical and logical power of the designer;
3. To relieve the designer from routine repetitious tasks.

Moreover, they suggest that the designer has three complementary tasks:

1. Control of the design process in information distribution;
2. Application of creativity, ingenuity, and experience;
3. Organisation of design information.

In the 1980's, several centres including the Bioengineering Centre at UCL decided the time was ripe to apply CAD techniques to prosthetics and orthotics. Lord and Jones in their discussion of issues in CAD for prosthetics and orthotics [LORD & JONES 88] recognised that an integrated CAD/CAM process would break down into three stages:
(a) Measurement whereby body shape and other pertinent information are converted to digital data;
(b) Shape generation by integration of individual measurements, a knowledge base and interactive adjustment via the computer;
(c) Manufacture of this physical component via a computer-controlled machine.

At UCL, the design and manufacture of prosthetic sockets for below-knee lower-limb amputees was chosen as the first project for the introduction of CAD, and it was broken down into stages following the pattern outlined by Lord and Jones. The crucial part of the system was regarded as the shape generation, stage (b), which involved the distortion of the original body shape to yield pressure in certain areas of the socket and relief in others, a process known as 'rectification'. Rectification templates are determined by a series of measurements made on the corresponding craft-based procedure. Stages (b) shape generation and (c) manufacture could be repeated for an iterative design of a final socket, if required. The result was a software package, Computer Aided Socket Design (CASD), for below-knee socket design, commercially available since 1989. At this time, the author joined the Bioengineering Centre. He was part of a small team responsible for the philosophy behind a CAD system for the design of above-knee sockets, and had sole responsibility for the development of the software for this second CASD module, commercially available since 1990, and which has been documented as Chapter 2 of this thesis. Other packages for the design of below-knee lower-limb prosthetic sockets also became available during the 1980's [SAUNDERS et al 85, HARLAN & BOONE 89].

The advantages offered by CAD systems to prosthetics and orthotics are several. One advantage is that by using computer graphics, the result of the design can be visualised before it is actually manufactured. This can mean that certain features in the design become modified before the item is produced, possibly reducing the amount of modification required after manufacture. Other analytical techniques, such as volume or mass analysis, can be carried out by the computer with similar benefits. A further advantage is that the computer can record exactly what operations and calculations are made to design the item.
Introduction

Therefore an exact replica can be easily produced for repeat orders. This is a feature which is commonly unavailable in prosthetics and orthotics since each item requires individual input from a skilled person whose decisions and actions are by their nature not recordable in precise numerical detail. Time is often a consideration in the design of a prosthetic or orthotic component. Often, CAD systems increase the time efficiency of the overall process by enabling the designer to concentrate on those stages where his experience and expertise are most required. Time and efficiency improvements are anticipated within a prosthetic and orthotic environment where a good CAD system is employed. Because the design of any custom-made item is an iterative process, especially if the criteria affecting the design alter with time, then since all the previous designs for a certain patient can be recorded, the iterative process can be easily monitored to the benefit of the final design. This is perhaps the most significant advantage of a CAD system in this setting, but each of these advantages has been demonstrated with the CASD system.

The CASD package developed at UCL and several other packages previously mentioned produce socket shapes which are defined by the positions of a number of discrete points on the surface shape. Such a method of shape definition implies that the precise shape of the socket between the known discrete points is not determined, and, in practice, it is the manufacturing technique which passively ensures the smoothness of the component to a large degree. The surface modelling work in the manufacturing engineering sectors by Coons, Bezier, Faux and Pratt and others mentioned above and summarised by Barnhill and Böhm et al [BARNHILL 83, BOHM et al 84], provides surfaces whose shape is entirely defined, and this fact implies that smooth surfaces can be guaranteed. Other advantages of surface modelling are that storage efficiency may be enhanced by requiring less information, and controlled adjustments of the surface are more easily quantified. One motivation for the current project was to add these techniques to the CASD package to determine the advantages which they offer. To date the possibility of advantage has been suggested [REYNOLDS 88, LORD & JONES 88], but not implemented or evaluated.

Some surface modelling techniques, for example B-splines, bi-beta functions and Bezier patches have previously been used in prosthetic situations [WALKER 88,
VERGEEST et al 87, SAUNDERS et al 85], but the implementation of these schemes requires a rectilinear grid of points for the model, and this has restricted their application to cases where a distorted sheet with a rectangular mesh can be wrapped over the surface. This is applicable in the approximately cylindrical case of prosthetic socket design, but for other cases such as orthopaedic footwear or the orthotic insole, the situation is inappropriate, leading, for example, to consequent loss in smoothness of the final design. Moreover, CAD techniques available to date for prosthetic and orthotic applications are each dedicated to one particular situation. A second motivation was therefore to further develop the surface modelling techniques used in engineering applications to generate a CAD philosophy and system which would be applicable in many prosthetic and orthotic instances. The author was the research assistant from 1987 to 1989 for a SERC funded project at University College London looking into surface representation possibilities for orthopaedic footwear, and application of the system will be demonstrated in work arising from that project.

The UCL CASD system is regarded as a First Generation system because of the limitations suggested in the previous two paragraphs - first, the system at no stage has a knowledge of the entire surface shape which it designs, and second, because of its reliance upon the structure of the data being in a rectangular grid, the philosophy cannot readily be extended to a general system for prosthetics and orthotics. The aim of the current work is to lay the foundations for a Second Generation CAD system for prosthetics and orthotics by the inclusion of surface modelling, so that the final surface could be completely controlled, and that the system should be readily adaptable for many situations in these fields. In the manner of Lord and Jones, the Second Generation system would have constituent stages as shown in Figure 1.1:-

(i) Measurement or capture of the body shape and input of other pertinent information;
(ii) Examination and checking of the measurement data;
(iii) Construction of a surface model for the particular situation under consideration;
(iv) Adaptation of the measurement data to become the data required by the surface model;
Figure 1.1 The constituent parts of an integrated CAD/CAM system for prosthetics and orthotics.

(v) Shape generation by integration of individual measurement, a knowledge base, the surface model and interactive graphical adjustment;

(vi) Output of the shape information for manufacture of the physical component under computer control;

(vii) Iterative design and manufacture of an improved component based upon clinical assessment, including patient comfort and other factors.

Each of these stages is now briefly discussed with regard to the envisaged system.

Measurement Techniques for measurement, or ‘Data Capture’, are not central to this thesis. Each technique involves significant development and experience of its own. However, it is important to understand the structure of the data which the techniques yield. In particular, the system must be able to handle the data from several measurement techniques and it would be useful to know if the data structure could be beneficial to the surface modelling.
Examination of measurement data Often measurement techniques will have their own facilities for checking the measurement data. However, some may not have, and the system should incorporate a software tool which allows visual presentation of the measured data together with the ability to edit the data, thereby removing erroneous points, for example.

Surface Model Construction At some stage, the surface model appropriate to any situation will need to be constructed. It is likely that the end user, however, will not want to be involved with the technicalities of surface modelling, and so the system will contain a library of models for the cases to which the system will be applied, and the user will then choose the model appropriate to his situation.

Adaption of Measurement Data After choice of the relevant surface model, the system will adapt the measurement data so that it is in the correct form for the surface model.

Shape Generation The system will generate an initial shape by replication of the measurement shape, and the system will give a visual presentation of this shape. The adopted shape will be formed by the integration of the measurement shape and its graphical portrayal, a knowledge base which contains an understanding of how the shape is affected in the particular situation under consideration, and interactive adjustment in response to both the image presented and patient or other comments on previous designs.

Output for Manufacture Although the manufacture of the component is not part of the current work, the system must produce shape information which is readily adaptable to various computer controlled manufacture processes. With surface modelling techniques it will be possible to produce data in the form of positional information on the surface at any required density and in any required pattern.

Iterative Design Design of orthotic and prosthetic components is rarely perfect at its first attempt. Often, comments upon and experience of previous designs can improve the result, and therefore the system must allow for iterative design.
Information from previous designs should be stored by the system and introduced at the stage of shape generation, with the prosthethist or orthotist controlling the reaction to this information.

There is no reason why several of the stages should not be remote from the others. This would open up the possibility of having one central design and manufacture site serving either many outlying measurement sites to which the manufactured items would be returned, or one mobile measurement facility which would tour the outlying product destinations. If the elements of the process are expensive, this could ensure a significant increase in cost-efficiency.

A general CAD system for prosthetics and orthotics is not envisaged to replace the trained prosthethist or orthotist, but rather to be an aid to him in consistency and efficiency. In the manner of Besant and Lui, the function of the CAD system is envisaged to be:-

(i) To serve as an extension to the memory of the user, especially as an aid to the iterative process of design;
(ii) To enhance the analytical and deductive power of the user;
(iii) To carry out standard rectification, modification or other design procedures for the user, i.e. to act as an expert system;
(iv) To allow visualisation of the designed item before manufacture.

The responsibilities of the trained prosthethist or orthotist will be:-

(i) Overall control of the design process;
(ii) Interpretation of the clinical requirements into quantitative inputs for the system;
(iii) Interpretation and action upon the results of the analysis and visualisation;
(iv) Organisation of the design information, including information such as patient comment on previous designs.
1.2 Contributions of the Current Study

In this section, the contributions of the current project are summarised.

(a) The extension of the UCL Computer Aided Socket Design (CASD) system to above-knee prostheses is described and documented. The presentation given here is complementary to the earlier discussion by Reynolds and Reynolds and Lord of the UCL CASD system for below-knee prostheses [REYNOLDS 88, REYNOLDS & LORD ip];

(b) A software program for the evaluation and examination of three dimensional data files is presented. The program has the useful features of allowing the graphical spatial editing of points in the file and the addition of marker points to the file;

(c) A review of data capture techniques used in prosthetics and orthotics is presented with the form of the data file produced of particular interest;

(d) A review of curve and surface modelling techniques is presented with particular emphasis on their use to interpolate a number of data points;

(e) A new surface model which smoothly interpolates a number of given (x,y,z) points is developed. Explicit conditions and procedures for the construction of the surface are given together with methods for modification of the model;

(f) Methods are presented for adapting the new surface model to particular situations and fitting data files to the model with specific examples given for the prosthetic socket, the whole foot and the orthotic insole;
Surface modelling is introduced to the UCL CASD system to determine and attempt to quantify the advantages which surface modelling offers over a system which contains a knowledge of the surface only at discrete intervals;

Initial experimental results are presented to determine how well the new surface model fits data sets from the whole foot and orthotic insole applications;

A surface visualisation software program is presented, which includes the ability to display objects comprised of the new surface model in both wire frame and shaded solid modes.

1.3 Structure of the Current Thesis

In this chapter, the term Computer Aided Design has been introduced together with its brief history, its common uses and the potential benefits it offers to prosthetics and orthotics. In Chapter 2, the system developed at UCL to automate the design of above-knee prosthetic sockets is described. This system itself is not the major contribution of the thesis, but rather, as previously outlined, is regarded as a springboard for the development of a more general CAD system for prosthetic and orthotic application to which the remainder of the thesis is dedicated.

Applications of a CAD system in prosthetics and orthotics involve the capture of an existing shape, often part of the human anatomy. Therefore, Chapter 3 presents a review of data capture techniques, including tactile, visual and other methods, and suggests the structure and form of the data which should be expected by a CAD system. After data have been captured from whatever device is appropriate, there is often a need to examine and edit the data. Moreover, it was found in the implementation of the developed surface model that marker points to highlight features such as bony prominences were frequently useful. Chapter 4 presents a software tool which enables graphical examination, editing and addition of marker points.
For the system to produce a smooth surface, an understanding of the mathematical definition of the surface designed will be required. Chapter 5 presents a review of curve and surface modelling techniques, with a particular emphasis on how these techniques are used to construct surfaces which interpolate a number of given data points. A conclusion from Chapter 5 is that further development is required to derive a smooth surface model which will cater for many of the shapes commonly encountered in prosthetics and orthotics. This development is the subject of Chapter 6, the result being a generally applicable surface model. Even after the addition of marker points and the examination of the data file, the data are not in the right form for the surface model. Chapter 7 presents a method for adapting the surface model to particular situations, and gives examples of the use of differing techniques for the fitting of data sets to the new surface model in these situations. Examples are given from prosthetics and orthotics. As yet, the advantages to prosthetics and orthotics of a smooth surface model over a technique which involves only discrete positional information about the surface are suggested but unproven. In Chapter 8, tests of these advantages are undertaken by application of the new surface model to the UCL CASD system. Other results are presented to determine the reliability of the fitting techniques of Chapter 7, and how well the surface model represents data in sample prosthetic and orthotic situations. In order to gain a good understanding of the shape of the surface before it is committed to a manufacturing process, it is necessary to visualise the surface, and Chapter 9 presents the surface visualisation tool developed during this project.

In Chapter 10 the main findings of the current project are summarised and suggestions are made for future work.
Chapter 2

A First Generation CAD System -
The UCL Computer Aided Socket Design System

2.1 Introduction

2.2 Conventional above-knee socket Design

2.3 Philosophy of the system

2.4 Brim shapes and sizes

2.5 Capture and storage of residual limb shape

2.6 Blending brim and residual limb shapes

2.7 Shape visualisation

2.8 Modification of the socket shape

2.9 Shape carving and socket production

2.10 The user interface

2.11 Remarks
2.1 Introduction

The total lower limb amputee population in England in a recent survey was 51,130, or 0.1% of the general population [DHSS 86a, DHSS 86b]. Following work to improve consistency and productivity in the manufacture of the prosthetic sockets required by this population, the Bioengineering Centre of the Mechanical Engineering Department, University College London developed its own CAD system for custom-designing below-knee prosthetic sockets. The system has been extensively documented elsewhere [REYNOLDS 88, DEWAR et al 85, DEWAR & REYNOLDS 86].

After the success of the below-knee prosthetic socket CAD system, currently being sold in the UK, the USA and elsewhere, the Bioengineering Centre developed its own CAD system for custom-designing above-knee prosthetic sockets. The author was involved in this project from the early stages and was solely responsible for the brim shape and size, blending, rectification and user interface software, and contributed to the overall system design as part of the small research team. The CAD system is named Computer Aided Socket Design (CASD).

The conventional above-knee prosthetic socket design technique is first described in this chapter because the chosen philosophy for the UCL CASD system is to reproduce as closely as possible the design method used conventionally. Some advantages of such a philosophy are that each stage of CASD can be compared with its respective stage in conventional design, and also that such a CAD system will be more easily accepted into the prosthetics field by operators who are naturally wary of computers. The reasons behind the choice of this philosophy are described later in the chapter.

For the purposes of designing a CAD system, the process may be separated into three stages:-(i) the capture and storage of the shape of the patient's residual limb, (ii) the design of a good socket shape, and (iii) the automated socket manufacture.
The parts of the system of central concern to this chapter are those concerned with design of the socket shape rather than the shape capture and the manufacturing process. The shape capturing technique used in the UCL system is described concisely here, but it is the same as that used in the below-knee prosthetic socket system, which is more completely documented by Reynolds [REYNOLDS 88]. Many commercially available computer-controlled machines are capable of producing a physical positive shape; more efficient socket manufacture may be possible with purpose-built machinery. The manufacturing system used at the Bioengineering Centre is briefly described.

2.2 Conventional Above-Knee Prosthetic Socket Design

A prosthetic socket is the interface between an amputee and his artificial limb. His comfort while wearing this component is a major factor in determining his 'quality of life'.
Socket design is specific to the user and it must achieve a close fit while providing stable support of the residual limb through those areas best suited to tolerate loading. The socket should be comfortable and stable and allow dynamic movement such as walking. The shape best able to satisfy these conditions is not a simple copy of the residual limb as this, for example, would cause loading in the wrong areas such as the distal end of the limb. Rather, distortions are incorporated into the limb contours which judiciously encourage loading in certain areas and relief in others.

The design and manufacture of an above-knee prosthetic socket can be divided into five stages:- brim fitting, casting of the residual limb, manufacture of a plaster positive from this cast, rectification and final manufacture of the socket.

The first stage of the process to design the correct shape for an above-knee prosthetic socket is brim-fitting. This follows a ‘quadrilateral’ socket philosophy suggested by Foort [FOORT 63]. Typically a prosthetist will have one or more sets of brims which are basically scaled versions of each other to accommodate a large range of patient sizes. Examples of commonly used brims are the Hosmer
brims and the Blatchford European brims, Figures 2.1 and 2.2. The Hosmer brims typically come in sizes $5\frac{9}{16}$", $5\frac{7}{8}$", $6\frac{3}{16}$", $6\frac{2}{16}$", and $7\frac{1}{8}$", and are adjustable at the medial and lateral extremes. The Blatchford European brims typically come in sizes 40 cm, 42 cm, 44 cm, 46 cm, 48 cm, 50 cm, and 52 cm, and are not adjustable. The shape of the brim will determine the shape of the proximal portion of the socket.

The prosthetist measures some key dimensions on the residual limb, namely:

(i) the antero-posterior dimension from the top of the adductor longus tendon to the ischial tuberoscity (AP dimension), Figure 2.3, (ii) the medio-lateral dimension from the head of the trochanter to the ischial tuberoscity (ML dimension), Figure 2.4, and (iii) the circumference at the perineum, Figure 2.5. Often, a prosthetist will reduce the circumference dimension by approximately 6% so that the final socket will fit snugly in this area. These dimensions indicate to the prosthetist which brim will fit the patient.

For a quadrilateral adjustable brim, such as the Hosmer brim, the ML dimension should correspond approximately with the size of the brim chosen. The medial
Figure 2.4 Measuring the medio-lateral (ML) dimension.

Figure 2.5 Measuring the circumference (Circ) dimension.
and lateral openings are then adjusted so that the AP and Circumference dimensions correspond to measurements on the brim, Figure 2.6. The brim is mounted on a jig in which the patient stands, and any necessary fine adjustments are made so that the patient feels comfortable. For a non-adjustable brim, such as Blatchford's European brim, the circumference dimension determines which brim is chosen, and the brim which has circumference value nearest to the value measured on the patient is fitted to his leg.

Prior to casting, the residual limb is covered with a thin nylon sock to enable the cast to be removed easily, and the limb is put into approximately $10^\circ$ flexion since this is roughly the position the limb would assume without loading. While the patient is standing in the brim, the prosthetist takes a cast of that portion of the residual limb which protrudes. This is done by wrapping plaster of paris bandages over the limb and the sock while the brim is weight-bearing, and extending the bandages up to cover the bottom edge of the brim. When fully set, the cast is removed.
A positive model is then prepared by pouring plaster mix into the wrap cast. It is usually at this stage that an axis is established within the prosthetic shape. Before filling with the plaster mix, a mandrel is positioned within the wrap cast. The orientation of the prosthetic shape relative to the mandrel is maintained throughout rectification and socket fabrication and eventually fixes the 'bench' alignment which determines the initial set-up of the socket on the artificial limb. The wrap cast is cut away to leave a plaster positive.

At this stage the socket is ready for rectification, a sculpting process in which material is removed in those areas where pressure is to be applied and added where relief is necessary. Often a depression is introduced at the proximal end of the positive to lock the trochanter into position and prevent the final socket rotating. Other rectifications may include relief to prevent irritation caused by rubbing under the ramus.

Final manufacture of the hard shell socket is commonly achieved in England by drapeforming by hand or with a semi-automatic RAPIDFORM machine. A nylon sock is put on the rectified plaster positive before a pre-heated polypropylene sheet is drawn down to form the socket shell. Vacuum is applied to encourage conformity between all surfaces. Once the polypropylene has cooled, the plaster positive is broken out and residual material is trimmed away to complete socket fabrication.

The fit of the socket will be assessed over a period of time as the patient wears it. If there are minor changes which would improve the fit, extra padding can be added internally to the socket, or the rim can be trimmed. However, if more fundamental alterations are required, the entire procedure must be repeated since the plaster positive was destroyed on breaking it out of the polypropylene socket.
2.3 Philosophy of the System

Prior to the development of the UCL CASD system for above-knee amputees, there was no commercially available system regularly used in the fitting of such patients. Therefore it is informative to look at the philosophies employed in CASD systems for below-knee amputees.

Saunders and Foort of UBC, Vancouver had produced a computer based system for designing lower-limb sockets [SAUNDERS et al 85] before the UCL system was developed. The UBC system uses a limited number of limb measurements to scale a ‘reference’ socket shape to provide a reasonable socket design. Thereafter, interactive on-screen modifications are made to the shape to produce the final fit. Sets of measurements are taken with calipers and tape-measure. Antero-posterior and medio-lateral measurements at the knee are used by the software to scale the proximal part of the reference socket in cross-section and length before antero-posterior and circumferential measurements distal of the knee joint, at one inch intervals, are used to ‘taper’ the scaled reference shape. Final modifications, where patches of material are added or removed to improve fit, are made via an interactive graphics display which can produce wire frame models and plots of cross-sections.

In numerical terms, the ‘scaling’ and ‘tapering’ operations are transformations of the reference shape which is stored as a set of cylindrical polar coordinates. The patches are added to the transformed shape by superimposing displacements onto the coordinate data. These patches are generated using bi-beta functions with zero value and derivative at the boundaries. The philosophy of this system is that all limbs have broadly similar structure, and are therefore accommodated by sockets of broadly similar shape and that the full ‘family’ of these shapes may be produced by varying a few geometric parameters on a ‘reference’ shape.

An alternative philosophy for below-knee CASD is to accept conventional practices and automate the measurement and rectification processes. In this case
measurement of the full limb surface is necessary. Rectifications must be correctly positioned according to each amputee’s anatomy or, if legitimate, in standardised locations relative to a number of reference points.

This latter philosophy was favoured by UCL since early versions of the UBC software were not flexible enough to produce sockets for a wide range of limb types without a large amount of interactive modification, and further scaling parameters were desired to extend the ‘family’ of available socket shapes. Therefore, for the UCL system, reproduction of conventional techniques was considered more likely to produce a successful fit since the underlying anatomical shape is present in each socket and the rectifications are relatively small distortions of this shape.

A CASD method for the design of above-knee prosthetic sockets based on a ‘family’ of shapes has been published [KROUSKOP et al 87, KROUSKOP et al 89]. However, the philosophy of this system is that used by Saunders et al for the below-knee socket design, and is open to the same limitations. A similar philosophy is suggested by Young [YOUNG 84], although that project was based on two unreliable measurements, and the report does not detail any patient fittings.

As with the below-knee CASD system, UCL favoured the philosophy of mimicking conventional techniques. This involves fitting a brim shape to the patient before casting and rectification, and is described more fully in the previous section. It was envisaged that the shape capture, small scale modifications and socket manufacture would be the same as those employed in the below-knee system. In some sense, this philosophy is a cross between the two which emerged for below-knee socket design: the proximal end of the socket is originally one of a ‘family’ of brim shapes, and the distal end is based on the patient’s individual measurements. As mentioned in the previous section, this philosophy has advantageous side-effects in that each stage in the process can be compared with the respective conventional stage, and the system is more readily accepted by computer-wary prosthetists.
Boone and Harlan, whose CASD system Shapemaker [HARLAN & BOONE 89] is widely used for below-knee prosthetic socket design, claim that their system is applicable for above-knee socket design. While this is undoubtedly true, it is difficult to see how the contours of an existing brim shape would be matched to a patient’s individual measurements.

Considerations of shape representation and adjustment led to the conclusion that it would be best to adopt a similar approach to the UCL below-knee system. The shapes are represented by a large number of discrete surface coordinates which locate nodes at the corners of facets to comprise a surface mesh. It would be preferable to have a continuous mathematically-defined function determined by a number of parameters to approximate the surface within a given tolerance. Then surface continuity would be guaranteed rather than having to be checked for point by point. However, this is a fundamental development in the CASD system, and is one aim of the Second Generation system developed later in this thesis.

### 2.4 Brim Shapes and Sizes

Fitting a brim shape of the correct size is the first stage in conventional socket manufacture. In an analogous way, the starting point for the UCL CASD system is to decide on the correct brim shape and size and store that within the computer.

It was observed that different sizes of the same ‘family’ of brims used in conventional manufacture are essentially scaled versions of one another. A plaster cast of one size brim was made and its shape digitised and stored in the computer. Since the different sizes are scaled versions of each other, the computer need only store one brim of each ‘family’ which the prosthetist would normally use.

Since a prosthetist can normally fit a correct size brim from a few dimensions, it should be possible to scale the brim size by using only the same few dimensions. However, it was necessary to quantify exactly how these dimensions relate to the setting of the brim used. When prosthetists at
Roehampton fit a patient, the intention is that the patient should have his ischium resting on the posterior shelf approximately 10 mm onto the shelf, the adductor longus should fit into the anterior-medial corner of the brim, and the trochanter head should be contained within the lateral wall of the brim, Figure 2.7. If the brim is adjustable these measurements then relate to the dimensions of the brim shown in Figure 2.6. A number of patients were measured and brims fitted to them in the conventional manner. The correspondence of the measured dimensions and the respective dimensions of the brim is given in Table 2.1. It can be seen that the dimensions of the brim are correct to within the accuracy of measurements taken with calipers and tape-measure, and that the Circ dimension is particularly difficult to measure accurately.

Thus when using the CASD system, the prosthetist measures the same three dimensions as he would conventionally; namely an AP dimension, an ML dimension and a Circumference dimension as described in section 2.2. He is asked to input these dimensions into the computer when the wrap cast is measured. If the user declines to enter the AP and ML dimension, but only enters
Table 2.1 Comparisons between AP, ML and Circ dimensions in millimetres taken from a patient's limb, and the corresponding measurements on a brim adjusted to fit comfortably, measured according to Figure 2.6.

<table>
<thead>
<tr>
<th>Patient</th>
<th>AP</th>
<th>AP</th>
<th>ML</th>
<th>ML</th>
<th>Circ</th>
<th>Circ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>limb</td>
<td>brim</td>
<td>limb</td>
<td>brim</td>
<td>limb</td>
<td>brim</td>
</tr>
<tr>
<td>SAD</td>
<td>95</td>
<td>95</td>
<td>155</td>
<td>140</td>
<td>455</td>
<td>430</td>
</tr>
<tr>
<td>FAR</td>
<td>95</td>
<td>95</td>
<td>150</td>
<td>150</td>
<td>445</td>
<td>460</td>
</tr>
<tr>
<td>HAS</td>
<td>100</td>
<td>100</td>
<td>145</td>
<td>150</td>
<td>430</td>
<td>450</td>
</tr>
<tr>
<td>UTT</td>
<td>90</td>
<td>95</td>
<td>130</td>
<td>135</td>
<td>390</td>
<td>405</td>
</tr>
</tbody>
</table>

The problem of fitting a correctly sized brim to a patient is now reduced to scaling the standard brim of a chosen family to match the observed dimensions. To quantify this process, for the quadrilateral brim used at the Bioengineering Centre, the following analysis was made:-

\[ \text{AP Dimension} = \frac{1}{5} \text{Circumference Dimension} \]

\[ \text{ML Dimension} = \frac{1}{3} \text{Circumference Dimension} \]

Let the standard brim have size \( S \), say. It may be convenient to set \( S = 1.0 \), but not essential. Then open the medial and lateral openings of the brim so that it has Circumference Dimension \( C \), AP Dimension \( \frac{1}{5}C \) and ML Dimension \( \frac{1}{3}C \). There are only two adjustable components - the medial and lateral openings - and three dimensions to be fixed with one degree of freedom. This observation means that the system is exactly determined and there is only one possible setting of the openings to achieve the result. Let this set up have medial opening \( M \) and lateral opening \( L \), Figure 2.8.
We also observe that as the medial opening of the brim is increased by an amount $\delta$, the AP, ML and Circumference dimensions are increased by $p\delta$, $0$ and $\delta$ respectively, where $p$ was found to be 87% experimentally. As the lateral opening of the brim is increased by $\epsilon$, the AP, ML and Circumference dimensions are increased by $q\epsilon$, $0$ and $\epsilon$ respectively. Note that $p$ and $q$ sum to 1:

$$p + q = 1$$

Suppose that we require a brim with dimensions $AP$, $ML$ and $Circ$. To achieve this the standard brim will be scaled to size $S'$ and have medial and lateral openings $M'$ and $L'$ respectively. An exact scaling of the brim to size $S'$ set up with $\frac{1}{3'}$, ratio will have all of the dimensions and openings of the standard brim multiplied by a factor $\frac{S'}{S}$. This fact, and the way the dimensions vary with respect to the openings give the following three equations:-

$$\left(\frac{S'}{S}\right) = \frac{ML}{\frac{1}{3}C}$$
These equations can be solved for the quantities which are being sought:

\[ S' = \left( \frac{ML}{\frac{1}{3}C} \right) S \]

\[ M' = \left( \frac{S'}{S} \right) M + \frac{1}{(p-q)} \left\{ AP - \left( \frac{1}{5} - q \right)\left( \frac{S'}{S} \right) C - qCirc \right\} \]

\[ L' = \left( \frac{S'}{S} \right) L - \left( \text{Circ} - \left( \frac{S'}{S} \right) C \right) - \left( M' - \left( \frac{S'}{S} \right) M \right) \]

The adjustable-type brim used at the Bioengineering Centre had values:

\[ S = 1.0 \]

\[ C = 450\text{mm} \]

\[ M = 12\text{mm} \]

\[ L = 37\text{mm} \]

\[ p = 0.87 \]

\[ q = 0.13 \]

So in this instance, the equations can be simplified to read:
These give the scaling parameter for the standard brim, and the medial and lateral openings for the scaled brim to fit the dimensions taken from the patient. Trials of this method of brim size adjustment have indeed been successful when fitted to the patient, showing that the analysis is valid, and this first stage of socket design is complete.

For a non-adjustable brim, for which the standard brim has circumference $D$ at the Perineum level, and for which a circumference dimension $Circ$ is required, the whole brim is scaled by the factor $\frac{Circ}{D}$. A brim with $D = 500\text{mm}$ was successfully fitted to patients at the Bioengineering Centre.

### 2.5 Capture and Storage of Residual Limb Shape

For the distal portion of socket design, the philosophy adopted for the UCL CASD system is based on measured external shape of the limb. This requires a method for capturing this shape data with sufficient accuracy in a reasonable time and storing the data in a convenient format. As with the below-knee CASD system, it was considered advisable to capture shape information from a plaster cast of that portion of the limb which is of interest. This is the conventional method for obtaining a measurement from the patient, and so is well-known to each prosthetist, it is inexpensive, and has the hidden benefit of ensuring the final volume of this portion is correct irrespective of the pressures caused by the prosthetist's hands. Krouskop et al have suggested an alternative method for shape capture, without these advantages, and which requires the patient to remain still for up to seven minutes [KROUSKOP et al 87, KROUSKOP et al 88].
Since the brim shape is used for the proximal portion of the socket, it is not necessary to store the shape of the entire residual limb. It is, however, necessary to decide in a repeatable manner how much of the limb needs to be digitised. The posterior-medial portion of the brims used at the bioengineering Centre were replicated for this purpose. One of these is illustrated in Figure 2.9.

Before the casting procedure commences, the residual limb is fitted with a nylon sock to avoid discomfort during casting. To establish the limits required for the casting, the posterior-medial portion of the relevant brim is offered up to the patient, so that his ischium is resting on the posterior shelf as when sitting in a standard brim on a jig. The prosthetist then traces a horizontal line with an indelible pencil onto the nylon sock, and marks off a vertical tick at the point which he judges to be the centre of the posterior aspect of the limb, Figure 2.10. The marks made with indelible pencil are transferred to the cast, and later used to align the cast in the correct orientation. The prosthetist casts up to about half an inch above the horizontal line he has just marked. After the cast has set and is removed, it is ready for digitisation.
Prosthetic sockets and residual leg shapes have cylindrical topography and it is convenient to take measurements with shapes rotated about their long axis. A regular mesh of surface coordinates can be established by measuring at fixed intervals of both angle and axial displacement. In the UCL CASD system, prosthetic surfaces are represented by a regular grid of radial measurements in slices around a central axis from the distal to the proximal end. This gives a compact method of storing the data. Figure 2.11 shows a measurement grid, and illustrates a ‘slice’ and a ‘strip’ of data.

The number of discrete measurements needed to define a surface to a given accuracy depends to a large extent upon the ‘shape’ of the surface. There is always a trade off between the fact that larger intervals between discrete data points yield less accuracy in the surface between those points, and smaller intervals require a larger amount of data. There is a limit on the size of a data set which can be handled by a computer, but even below that limit, the more data, the more time it will take to process the data. The amount of time taken to process the data is approximately inversely proportional to the square of the
interval between data points. In the early Vancouver system, angular intervals of 5° and axial displacements of $\frac{1}{4}$″ were found satisfactory. The UCL CASD system for below-knee amputees was found sufficient with 10° and $\frac{1}{4}$″. This latter resolution is the minimum required for above-knee sockets, and it has been suggested that 5° and $\frac{1}{8}$″ would be preferable, although trials are currently being conducted on the coarser resolution. A typical size for a data file is 2000 radial values, which when stored in an ASCII data file, are approximately 15kBytes in size. This file is called a 'Measurement' file since it contains a measurement of the residual limb.

The apparatus for digitising the cast has been described in detail elsewhere [REYNOLDS 88, DEWAR et al 85]. Briefly, however, the cast is mounted on a rotating wheel, above which an arm moves a plane through the axis of rotation. Using a potentiometer in its base, the arm reads in the radial position of its tip relative to the axis. As the wheel is rotated, the arm travels along a leadscrew,
Figure 2.12 The measurement system.

traversing the length of the cast, while values of the radius from the axis of rotation are read in at regular intervals. A diagram of the measurement system is given in Figure 2.12.

The radial data initially lie on a helical path around the cast. These values are then transformed using linear interpolation into regularly spaced data points on slices and strips about the central axis.

2.6 Blending Brim and Residual Limb Shapes

At this stage, the computer has a knowledge of a brim shape and the shape of the distal part of the residual limb. These two files have been digitised at differing times, and are likely to have different alignments, and maybe different axes. Moreover, if the two files were correctly aligned and concatenated, they would not fit smoothly together. The files must therefore be aligned and then blended together before an acceptable shape for a socket is reached.
The first stage is to align the two shapes on one central axis, since although both files contain radial values relative to some vertical, the axis might not be the same vertical axis for the two files, and moreover, one file might be twisted relative to the other. The point which was marked on the residual limb when casting is already stored in the measurement file, and the same point is also in the brim file. Using these points, the brim file is rotated so that it is in the same orientation as the measurement file - that is so that the medial sides of the two data files will be facing in the same direction, and similarly for the posterior, lateral and anterior sides. Then the centre of the polygon defined by the data points of the top slice of the measurement file is calculated together with the centre of the bottom slice of the brim. These give the amount by which the brim must be translated in order to be aligned with the measurement. After the translation, the points on the brim are recalculated about the axis of the measurement file extended proximally to pass through the brim. However, the points of the brim will not be spaced at equal angular intervals about this new axis, and so are recalculated using interpolation where necessary. The resulting two files can be concatenated into one file since both now have the same orientation and axis.

After the above alignment there will still be an apparent join between the brim and the measurement. This is due to the fact that the unconstrained tissue of the central portion of the limb which appears at the proximal end of the measurement file will not naturally be the same shape as the distal end of the brim file. Thus the brim and the measurement shapes need to be blended together.

Figure 2.13(a) shows the profile of a typical strip of data in the concatenated file, and highlights the transition between its two portions. Each point on the strip is a radial value from some central axis. The profile is blended by calculating the discontinuity at the join, and linearly adding an increasing proportion of this discontinuity over the ten slices below the join. The resulting profile of the strip is shown in Figure 2.13(b).

The data file now has no discontinuous joins, but has sharp edges, or discontinuity in its tangents. The file requires smoothing. This is done by going
Through each strip in turn, and replacing the radial value at each point with a weighted average of the radial values of its surrounding points. The most convenient way to describe such an average is by considering its 'mask'.

For the mask \(\left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right)\) if the (i-1)th, ith and (i+1)th points in the strip have radial values \(r_{i-1}, r_i,\) and \(r_{i+1}\), then the ith radial value is replaced by a radial value \(\frac{1}{4}r_{i-1} + \frac{1}{2}r_i + \frac{1}{4}r_{i+1}\). For the mask \(\left(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}\right)\) if the (i-2)th, (i-1)th, ith, (i+1)th and (i+2)th points in the strip have radial values \(r_{i-2}, r_{i-1}, r_i, r_{i+1}\) and \(r_{i+2}\), then the ith radial value is replaced by a radial value \(\frac{1}{5}r_{i-2} + \frac{1}{5}r_{i-1} + \frac{1}{5}r_i + \frac{1}{5}r_{i+1} + \frac{1}{5}r_{i+2}\). The effects of different masks are shown in Figure 2.14. The mask adopted for the CASD system was \(\left(\frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}\right)\) partly because of its simplicity, but mainly because it works well in practice. A second mask, \(\left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right)\), was used around each slice below the posterior shelf in the data file. This eliminated any isolated errors resulting from the digitisation.
The final result of the alignment, blending and smoothing is one data file which represents an above-knee socket shape. At this stage the socket may or may not require further modification, and it is appropriate to be visualised.

### 2.7 Shape Visualisation

A large set of numerical data, such as a file produced by the CASD system is difficult to interpret without the aid of appropriate tools. The most readily understandable tool is a graphical image on the computer which resembles a three dimensional object as the eye might see it. Other forms of graphical output such as wire frame models or plots of cross-sections can be misinterpreted because the depth information is difficult to understand, and so a solid three-dimensional view is preferable and it is this which is incorporated into the below-knee CASD software, and now the above-knee CASD software.
A point on the surface has a corresponding point on an adjacent strip and these two points have corresponding points on the adjacent slice. Four points together define a facet, as in Figure 2.15. The UCL CASD system decides which facets are visible from a given viewpoint, and draws them as a grey-scale image, the shade of each facet depending on its orientation with respect to a known light source. More precisely, the shade is determined by the scalar product of an approximate unit normal vector to the facet and the unit vector from the light source to the centre of the facet.

The visualisation routines for the above-knee software are, with one major exception, largely those used in the below-knee software which allow the position of the viewpoint to be altered and the scale of the image to be increased and decreased. The difference concerns the choice of a low resolution or high resolution viewing mode. When low resolution graphics is chosen, the CASD system displays one plane facet between each group of four adjacent data points. When high resolution is chosen, an interpolation algorithm is used to estimate the radial data at intermediate points, and four facets are drawn between each four adjacent data points. It is the interpolation algorithm which differs.
The interpolation algorithm used for high resolution is arbitrary, but was found to be quick and produce reliable results. It is a one directional algorithm, and establishes a cubic curve through four consecutive data points in a strip or slice of data. Thus for the points shown in Figure 2.16(a), the algorithm is used on the slices to yield the interpolatory data points in Figure 2.16(b), and then on strips, including an imaginary strip, to give the interpolatory data points in Figure 2.16(c). In detail the algorithm is as follows:

Suppose that four consecutive data points on one slice have radial values \( r_3, r_2, r_1 \) and \( r_3 \). Let these points have parameter values \( t = -3, -1, 1 \) and 3 respectively. The choice of parameter values is arbitrary, but ensures easy mathematics since the required value is the radius halfway between the middle two parameter values, -1 and 1, ie at 0. A cubic curve

\[
r(t) = ar^3 + br^2 + ct + d
\]
is established such that

\[ r(-3) = -27a + 9b - 3c + d = r_{-3} \]

\[ r(-1) = -a + b - c + d = r_{-1} \]

\[ r(1) = a + b + c + d = r_1 \]

\[ r(-3) = 27a + 9b + 3c + d = r_3 \]

The required interpolation value is \( r(0) (=d) \). Adding the first and fourth of these equations we get

\[ 18b + 2d = r_{-3} + r_3 \]

Adding the second and third of these equations gives

\[ 2b + 2d = r_{-1} + r_1 \]

On elimination of \( b \), these latter two equations yield

\[ d = -\frac{1}{16}r_{-3} + \frac{9}{16}r_{-1} + \frac{9}{16}r_1 - \frac{1}{16}r_3 \]

This value for \( d \) is an estimate for the intermediate radial value, and usually is a realistic estimate, and gives good images in high resolution. Close observation of the formula for \( d \), however, shows that \( d \) is not a convex combination of the radial values, (the formula involves some negative multiples) and can exceptionally lead to undesirable values. The condition for this to occur is that the difference \( |r_3 - r_1| \) is much greater than either of \( |r_1 - r_{-1}| \) and \( |r_{-1} - r_{-3}| \), and an example of the result in such a situation is given in Figure 2.17(a). This feature occurs in the above-knee CASD system just above and below the posterior shelf,
and it is successfully treated in the following manner: when the difference between two consecutive radial values is greater than a certain threshold, for example 5mm, d is set as in the following formula:

$$d = \frac{1}{2} r_{i-1} + \frac{1}{2} r_i$$

This procedure has been implemented within the CASD system, and the effect is demonstrated in Figure 2.17(b).

### 2.8 Modification of the Socket Shape

After blending the brim and residual limb shapes, the computer contains a file of radial values which define an above-knee socket shape. However, the prosthetist often wishes to modify this socket in various standard fashions. The UCL CASD system allows for six such modifications, and, in addition, on-screen sculpting. The six standard modifications are:-
• 1 Modification of the length of the socket;
• 2 Modification of the ML dimension for brim size;
• 3 Modification of the Circ dimension for brim size.
• 4 Modification of the AP dimension for brim size;
• 5 Introduction of Flexion or Extension;
• 6 Introduction of Abduction or Adduction;

For modification of the socket length, the only alterations allowed are changes by multiples of the separation distance between consecutive slices. Linear interpolation is used to stretch or reduce the distal portion of the brim by the required amount.

If a modification to one of the dimensions of the brim is desired, then the CASD system redesigns the socket from the beginning, using the modified values. However, if complete freedom in choice of these dimensions is allowed, it is feasible to design 'impossible' shapes. An example of this is due to a very small AP dimension, when a brim shape is chosen which would require a medial opening to be of negative size. The program identifies many of the cases of impossible brims, and informs the user, not allowing him to proceed. At this point in the program, an alternative value is chosen by the program unless the user desires to make his own correction. The alternative is reached by trying to change each of the brim specifications in turn until a 'possible' brim shape is achieved.

For Flexion/Extension and Abduction/Adduction one further concept is required. The above-knee socket inherits from the brim shape two 'shelves' which are horizontal flanges on the brim, as can be seen in Figures 2.1 and 2.2, and are apparent in the data file as large discontinuities of radial value between consecutive slices. These are a posterior shelf, on which the ischium sits, and much of the patient's weight is taken, and a medial shelf which must be kept
Figure 2.18 The planes for Flexion/Extension and Adduction/Adduction shown in the slice containing the reference point. The angle, $\theta$, of a point or strip relative to the reference point is also indicated.

clear of the ramus bone for comfort reasons. Flexion/Extension involves an alteration in the angle the medial shelf makes with the horizontal, and Abduction/Adduction involves an alteration in the angle the posterior shelf makes with the horizontal. Conventionally a prosthetist achieves this by adding and removing plaster to the plaster positive. In the CASD system, the data is altered in an analogous way.

The alteration of the angles of the posterior and medial shelves is a critical factor in determining the comfort of the final socket, and it is important that the correct features are rotated. For this reason there is a file associated with each brim shape which indicates the point on each strip which is immediately above the shelves, and one point which will act as the centre of any rotations in the Flexion/Extension and Abduction/Adduction planes.
The overall structure of the data files implies that there must be one radial axis which is internal to each of the slices of the file. This in turn implies that difficulties may arise if the whole portion of the socket above or below the shelf line is rotated with respect to the other portion since the central axis may not now lie within the whole shape. Indeed such a procedure would not mimic that followed by a prosthetist. He just angles the shelf and may alter prominent features above the shelf accordingly. Therefore it was decided to angle the shelves without a general rotation of the proximal portion. This was achieved by looking at each strip of the data file in turn, and determining by how much to shift the proximal portion of that slice in an axial direction. More explicitly, examining the slice which contains the reference point, Flexion/Extension was assumed to act in the plane which contains the central axis of the data file and the reference point, and Adduction/Abduction was assumed to act in the perpendicular plane through the axis, Figure 2.18. Consider the strip at angle $\theta$
to the strip containing the reference point. Each point above the shelf mark is examined, and the amount that point would move, \( \delta z \) in the axial direction as a result of rotation of Extension angle \( \phi \) and Adduction angle \( \psi \) is given by

\[
\delta z = r_{ref} \cos(\theta)\cos(\phi) + r_{ref} \sin(\theta)\cos(\psi)
\]

Figure 2.19 gives a diagramatic representation of the manner in which this axial movement for the points above the shelf in the strip of consideration is achieved. The five slices below the shelf are expanded so that the shelf gives rise to an instantaneous discontinuity in radial value. The position of the shelf is then moved in the axial direction by the amount calculated using the above formula. The five slices below the shelf are stretched using linear interpolation to fill the portion beneath the shelf, and the points above the shelf are each moved in the axial direction by the amount appropriate to that point using the formula above, replacing \( r_{ref} \) with \( r_i \), the radial value in slice \( i \). Finally linear interpolation is used to ensure that final radial values are given for points which lie in slices.

After modification, the socket may still not be the exact final shape which the prosthetist desires. For final adjustments which are often small scale, sculpting is provided. Sculpting is a process in which the user can vary the radial value at any point on the surface while at the same time having continually updated three-dimensional and cross-sectional views.

Thus the prosthetist is able to arrive at the socket shape he desires, down to small details of shape.

2.9 Shape Carving and Socket Production

The procedure for carving above-knee prosthetic sockets uses the software and hardware used to carve below-knee prosthetic sockets [REYNOLDS 88].

Briefly, a plaster blank is secured onto an axial carving machine. The blank is able to rotate about its axis, and the carving tool has two degrees of freedom - one along the length of the blank, and one in a direction radial to the axis of
rotation. The data which are stored in the computer in slices are translated into radial values lying on a helical path along the axis by using linear interpolation. The carving tool then travels the length of the blank while the blank rotates, and all the excess material is removed.

To produce a socket, the carved plaster model is draped with a pre-heated polypropylene sheet, and then vacuum is applied to pull the polypropylene into the correct shape. The plaster is broken out and the socket trimmed as for conventional design.

Work has been done to try to carve the socket directly [DEWAR et al 86, DEWAR & JONES 90], but the results are not yet reliable. Further development of the original mould shape, and computer control of the carving tool are required.

2.10 The User Interface

The interface between the CASD system and the user is via a mouse controlled menu driven program. The software is much the same as for the below-knee CASD system except for the instances listed below. These correspond to the stages in conventional design where the above-knee method differs from the below-knee method.

- (i) The below-knee System displays PATIENT, SOCKET, MEASUREMENT and RECTIFICATION cards. The above-knee System displays PATIENT, SOCKET, MEASUREMENT and BRIM cards.
- (ii) The BRIM cards describe the brim shapes available and describe them to the user. An application package has been written to allow redrawing of these cards.
- (iii) The MEASUREMENT cards show, in addition to the measurement information, the relevant dimensions taken from the patient.
- (iv) Error messages have been introduced when 'impossible' brims are specified.
2.11 Remarks

A CASD system for design of above-knee prosthetic sockets has been designed, and tried with eleven patients. The fittings on the patients were satisfactory, and good on ten of the eleven patients. Figure 2.20 shows a trial fitting of a patient wearing a socket designed by the CASD system. The one case which was unsatisfactory was when a brim style was tried on a patient which the prosthetist thought inappropriate anyway.

The system works, but has limitations. Never is a complete knowledge of the surface behaviour held, but only an understanding of the surface at a finite number of discrete points. Further information to shape may be included in a future development of a CASD system. Reynolds and Krous-kop et al have suggested that tissue stiffness should be quantified in some way and this description included in the design procedure [REYNOLDS 88, KROUSKOP]
et al. 89]. The CASD system is ideally suited to prosthetic sockets because of their basically cylindrical shape. However, it is not applicable to other shapes found in prosthetics and orthotics.

This study aims to develop a more generally applicable CAD system with a complete understanding of the surface involved.
Chapter 3

Data Capture Techniques

3.1 Introduction

3.2 Tactile Methods

3.3 Optical Methods

3.4 Ultrasonic and Other Radiation Methods

3.5 Remarks
3.1 Introduction

When using a Computer Aided Design (CAD) technique, in many applications such as building design or car design, the user has a large amount of freedom for the shape of the final object. Standard CAD packages give him the flexibility to meet the constraints he does have, while still employing his imagination and artistic flare in the overall design. In prosthetics, orthotics and many other medical applications, however, the user is constrained to design something which is a very close match to an existing object such as a foot or a residual leg after amputation. For this to be possible, the size and shape of the object must be determined in some digital form and input into the computer. In this chapter, the major techniques for data capture are reviewed, and their common applications are identified.

The data which are input into the computer are going to be used in a mathematical model of the surface of the object, and therefore it would be advantageous if the data might in some sense anticipate the structure of the information required in the surface model. Moreover, the structure of the data
which is passed to the computer is of particular importance to the CAD system
designed as part of this project, and so the main aim of the chapter is to
determine the form of data with which the CAD system must be able to cope.

Data capture methods can be split into three categories - tactile, optical and
radiation methods. A section of this chapter will be devoted to each one.
However, at this point, a tree of the various methods covered is presented,
Figure 3.1. There are a number of factors, such as cost, which may dictate the
use of certain methods, and these will be discussed in the remarks at the end of
the chapter.

3.2 Tactile Methods

Tactile methods, by their nature, require contact with the surface being digitised.
However, the parts of the human anatomy commonly occurring in prosthetics
and orthotics often have soft tissues just beneath the skin. Therefore, a tactile
method can disturb the surface shape by the pressure the contacting probe must
exert to determine that contact is indeed occurring. This effect is particularly
noticeable if the tactile method uses a probe or sensor moving over the whole
surface, but only digitising one point at any one moment. In this case, for
example, if a limb with an average diameter of 50 mm is being digitised, and
the pressure of the tactile sensor on the surface causes an average depression of
the tissue of 1 mm, then the volume of the digitised shape will be approximately
4% less than the volume of the limb. This is often an unacceptably high error.
Indeed, it is thought that accurate conservation of volume is important in the
design of prosthetic sockets for leg amputees. The method which the UCL CASD
system uses which does not suffer from this volume distortion is to take a plaster
cast of the limb, and to digitise the plaster cast. Because the cast is not flexible,
but rigid, the tactile probe is unable to deform the shape captured. It should be
noted, however, that the plaster cast technique was originally employed by the
UCL CASD system, not because of volume considerations, but merely for ease
of digitisation. Similar plaster cast techniques are commonplace in prosthetics
and orthotics, and can therefore be used to overcome the tactile contact problem.
Tactile methods can be divided into three basic types: hand-held sensors which are controlled by the user, mechanically held sensors which are controlled by a robot or other mechanical device and usually under the supervision of a computer with appropriate software, and specially developed arrays of sensors. These three types are now discussed. A general review of the sensors available is edited by Pugh [PUGH 86a], including examples of each type.

Hand-Held Sensors

The simplest digitisers consist of a hand-held stylus or wand, which is moved over the surface while capturing the coordinates of the tip of the stylus. Experience has been gained with two typical such stylus digitisers in the course of this work, the Perceptor [PERCEPT] and the 3Space Isotrak [ISOTRAK].

The Perceptor consists of a rigid base containing a microprocessor controller, from which a jointed arm with four rotation axes is mounted. Signals from a rotary potentiometer on each axis are fed into the microprocessor which then computes and outputs the coordinates of the distal tip of the arm. Because of the length of the arm, the range of the Perceptor is restricted to approximately 30 cm from a central position, with an accuracy of 1 mm.

The Isotrak is an electromagnetic device and consists of a source module, a sensor and a microprocessor controller. The source emits a magnetic radiation which the sensor, optionally in the shape of a stylus, detects. From signals fed back from the sensor, the controller again can compute and output the position at the tip of the stylus. The range of accuracy of the electromagnetic field detection is approximately a 2 m hemisphere on one side of the source, and an accuracy of 1 mm.

Both the Perceptor and Isotrak have similar operational modes of data acquisition - single point, free streaming or incrementally triggered. For this type of device, two methods of data capture can be envisaged. First, a set of
pre-defined single points could be digitised from the surface, or secondly, a larger number of points could be digitised during sweeps of the surface in continuous streaming or incrementally triggered modes.

The first method would require the user to decide beforehand exactly how many points on the object would be needed and their relative positions. The pen would be moved to each point in turn, and the coordinates of that point stored in a file. The strong advantage of this technique is that the points digitised could be exactly those, and only those, required for a mathematical model of the surface. For example, a rectangular array of points could be digitised on an object of appropriate shape.

For the second method, the pen can be used to run all over the object, continually sending data. The user ensures that the pen covers the entire surface at a reasonable density of points, so that no disc of radius $r$, say, is empty of data points. The order of the coverage of the surface can be controlled by the route which the pen takes, and so the data file will then contain points in the order...
desired. However, the exact number of points is difficult to control, and so the data file will be, in effect, a random covering of the whole surface. Figure 3.2 shows typical examples of data obtained by these two methods.

Mechanically-Held Sensors

Mechanically-held tactile sensors consist of a sensor on the end of a mechanical or robot arm, and supervisionary software to enable this arm to track over the surface of the object being digitised. These sensors are commonly found in marine situations where, due to the environment, visual methods are not applicable [HUGHES and BROOME 88, WRAY 87]. In marine situations, the sensors are often attached to expensive Remotely-Operated Vehicles (ROV’s), but in a prosthetic situation they would be constrained mechanically. If the shape of the object being digitised is of a known form, then this type of digitisation can be very effective. For example, the UCL CASD system involves the use of a digitiser which has a rigid arm, the tip of which is always in contact with the object being digitised, and angular potentiometers at the base of the arm which enable the position of the tip to be calculated. In this case the object is approximately cylindrical, and the tactile digitiser follows a spiral path around the object, while being mounted to a fixed jig. The result of this is a file of data points offset from a radial axis at regular angular spacing. This file is then translated to another file which contains slices of equally spaced radial data, to an accuracy of 1 mm [REYNOLDS 88]. If the shape of the object being digitised is not known, then for generality, a robot arm with a large number of degrees of freedom would be required, and this would be an expensive option. Complicated software would be required for the control of automatic data collection.

Arrays of Sensors

Most tactile systems applied in prosthetics and orthotics consist of one tactile sensor which gives information about one point at any one instance, but an alternative method of measurement is to use a specially designed array of
sensors to give more information at any one time. A simple example of this is the five-point array of sensors used by Greig et al [GREIG and BROOME 89], from which five positions are read at any one moment. These are used to calculate one position and one normal vector at that position. More intricate arrays of sensors are described in [PUGH 86a], but the ultimate 'cloth' which consists of a large array of sensors to be wrapped around the object under consideration is not available yet. This 'cloth' array would be able to digitise the whole surface at one time accurately and reliably.

3.3 Optical Methods

Optical methods for capturing the shape of an object are non-contact methods, and so do not require the same consideration as tactile techniques. However, they are reliant upon good lighting, and due consideration must be made before an optical method is chosen. In marine environments, the medium of sea-water can mean that optical methods can only work to a certain resolution, and other situations may require darkness in order to function properly. None of these hindrances occur in prosthetics and orthotics, and so optical methods are open to consideration. There are a number of different optical methods for capturing the shape of an object, but the principal ones are Moiré fringe techniques, photogrammetry, triangulation with structured light, and silhouette. These methods are described here.

Moiré Fringe Techniques

The father of Moiré fringe techniques is generally thought to be Foucault who in 1859 proposed the use of gratings of parallel light and dark strips for the testing of the quality of lenses. However, it was Lord Rayleigh who first observed the phenomenon of the fringes when he saw the pattern produced by two gratings at a slight angle in 1874. It was not until the 1950's that Moiré fringes were used to any great extent. Light passing through a grating produces light and dark lines when shone onto an object. If the light is passed through two gratings then fringes, known as Moiré fringes, are caused which are much more
Figure 3.3 The patterns from two gratings at an angle to one another form Moiré fringes.

apparent than the original light and dark lines. The basic principal is explained with the help of a diagram, Figure 3.3, where the Moiré fringes are indicated. If the two gratings are parallel, then the effect is that, on the object, light patches are apparent at certain depths, and dark patches at other depths. These light and dark patches can be used to give a contour map of the object being considered. The same effect can be obtained by shining the light onto the object through a grating, and viewing the object through the same grating at a slightly different angle. In 1969, Theocaris published a book on the use of Moiré fringes in the analysis of strain [THEOCARIS 69], and the reader is referred to this text for a greater analysis of the technique. A derivation of the fact that the light and dark patches occur at fixed depths is given by Duncan and Mair [DUNCAN and MAIR 83].

If a photograph is taken of a Moiré fringe pattern, then calculation can be made of the depths at points within the photograph. If the technique is used to give two or more photographs from different positions, then more precise positions can be calculated. Disadvantages of this technique are that for each position
required, the operator has to pick out the point in the photograph, and then determine from his knowledge of the shape of the object, the depth of the point. This is because there is no automatic way of telling from just one view whether the change from a light band to a dark band implies an increase or a decrease in depth of the point. Examples of Moiré fringes of human faces and bodies are given in [DUNCAN and MAIR 83]. The accuracy of the depth information can be increased by using a finer grating, but this will also increase the amount of information which must be deduced from knowledge of the shape. These problems can be overcome, especially for a convex surface, as demonstrated by Duncan and Mair in their reproduction of a foot last captured by Moiré fringe techniques.

Photogrammetry

'Photogrammetry' literally means the taking of measurements from photographs. The first users of this technique were Laussedat, a French army engineer, and Nader, a French photographer, who took aerial photographs from a balloon in the 1850's, although the technique did not become very advanced until sufficient photographic improvements were achieved. In the 1920's, American government agencies first used photogrammetry from aerial photographs to provide contour information in map compilation, and in the 1930's the art became much more widely used after a geological survey, again by American government agencies. Since then, the technique has become more widely used, although its commonest uses are in geology, archaeology and civil engineering in the study of the land [BURNSIDE 85, MOFFAT and MIKHAIL 80]. Successful applications to the medical field are by Vergeest et al and Burke [VERGEEST et al 87, BURKE 71].

Photogrammetry, or, more precisely, stereo photogrammetry, is a solution to the problem of parallax, and is similar to the system which our eyes and brains use to make sense of the three-dimensional world. As the name suggests, stereo photogrammetry involves the taking of two photographs of the same object, often with synchronised cameras, and always at slightly differing positions. The ratio between camera separation, \( b \) and the distance from the cameras to the
object, $d$, should be such that $0.3 < b/d < 1.0$. Figure 3.4. The directions of a point from two known places, where the camera was positioned, can be used to give a position of the point in three-dimensions. For derivation of the equations involved see [BURNSIDE 85] or [MOFFAT and MIKHAIL 80]. The calculation requires the two photographs to be mounted on a digitising tablet. The operator moves a measuring mark to the location of the point $r_i$ in each photograph, using $x$, $y$ and $z$ wheels and then for any point $r_i$ on the object, the coordinates $(x_i, y_i, z_i)$ can be determined. Overall scaling and alignment of the data is achieved by digitising a few points whose $(x,y,z)$ locations are previously known.

The operator has complete control over which points are digitised in photogrammetry, and this is a considerable advantage because if the operator knows in advance which points would be required for the surface model, he can ensure that only these points are digitised. However, photogrammetry requires
a large amount of operator time, where the operator is repeating a very sensitive and skilled task. Large errors can occur due to misplacement of the measuring mark.

**Triangulation with Structured Light**

A common technique for depth interrogation is to project a pattern of light onto the object, and to view the object from another angle. Triangulation can be used to determine the depth of points on the surface. Pugh considers that it is easier to use structured light such as a laser line and a mechanical scanning device to pass the line over the object being digitised, than to project a large pattern onto the object [PUGH 86b]. The scanning can be achieved either by moving the beam or moving the object through the beam. Information is taken from the position of the beam and its view at a number of discrete steps.

An example of the first method of projection of light and dark regions onto the object is the system designed by Quantec Image Processing Ltd [CURRAN and GROVES 90] for the measurement of human back shape. Here a pattern of parallel light and dark lines is projected onto a patient’s back. This pattern is then viewed from a different angle by a CCD camera, and one view, or ‘frame’, is captured, giving an image. From the triangle of viewpoint, image and light source, the three dimensional location of many points on each line are calculated. This process gives about 250,000 points for the digitisation of a human torso, with an accuracy for each point of ±2.5 mm.

An example of the method where a beam of light is panned over the entire object of interest is the Isis developed by Oxford Metrics [OXFORD], also used for scanning human back shape. For this application, the subject remains stationary while a horizontal light beam is swept vertically down over the surface which is essentially ‘fairly flat’. From the known distance between the camera and light source, and the angle of light projection, triangulation can calculate a series of points on the surface. The calculation continually passes along the projected beam of light as it moves continuously over the surface. Calculation starts from one extreme of the field of view, and passes along the light beam, determining
the position of a point on the light beam. When the calculation reaches the second extreme of the field of view, calculation immediately passes back to the first extreme, and the process continues. Meanwhile the beam is continually panning over the object. Figure 3.5 gives an example data file. The Isis claims an accuracy for each point of ±2 mm, and would give approximately 20,000 data points for a human back shape.

Lovell and Greig use this technique of passing a laser beam over an object for recognition and positioning of particular features on the surface under consideration [LOVELL 90, GREIG 91]. These applications are typical of the aircraft and marine environments respectively.

The alternative sweeping method consists of rotation of the object in front of a fixed camera and light source. Examples of this are the VideoLaser developed by 3D Vision [VISION] and a similar device developed by UCL’s Medical Physics Department [MOSS et al 88] for head scanning. In the latter, the object - a person - is seated on a chair which rotates in front of a fixed unit which
projects a vertical laser light line onto the midline of the head, the centre of which is aligned with the vertical axis of rotation. The camera views the projected line at specified angular intervals of rotation, and as a result the coordinate data file is structured onto a regular cylindrical mesh. Each surface coordinate lies on a radial spoke coming out from the vertical axis, with specified angular spacing between the spokes.

With a sweeping method where the camera and light source remain stationary, and the object rotates, the volume in which the projected line will lie is constrained to be the plane through the axis of rotation and the camera. This requires only a small amount of calibration compared with either moving the beam of light or projection of a large pattern. Therefore the accuracy of digitisation is considerably improved. With the UCL Medical Physics system accuracy of ±0.1 mm is claimed for the 10,000 points which would be collected for a human face.

Rioux has suggested a system where the scanner is synchronised with the laser to increase the resolution of the system, and because the angle between camera and light source is diminished, the problem of shadows is not so great [RIOUX 86]. He demonstrates his system by digitising one view of a shoe last and one view of a human hand.

When the method requires the object to be rotated in front of the camera, the object is restricted to being essentially cylindrical. The human face, as digitised by UCL Medical Physics, and the UCL CASD system are both situations in which the shape is appropriate.

Silhouetting

The final class of common techniques for optical methods of data capture is silhouetting. This technique is claimed to be used first by Dibutades in Ancient Greece. He is reputed to have carved a human form from shadows of the model projected at a number of angular directions. Silhouetting requires a good contrast between the ambient light and the object in view. Then thresholding of
the image can be used to obtain the silhouette. At a basic level, silhouetting has been implemented by Savut et al [SAVUT et al 89] to determine the size and location of a cylinder. A CCD camera is used to take an image of the object which is of a dark hue against a light background. The image is passed to a computer and analysed. The trace of the border between light and dark portions of the image determines a plane which is tangent to the object. Using several positions of the camera, a number of tangent planes are found. The possible cylinders which fit these tangent planes are worked out. For a sufficiently large number of tangent planes, there will be a unique cylinder which will fit.

One refinement for the edge detection technique is Sense Shape, being developed by Shape Products Ltd [SHAPE] for the digitisation of lower limb residua. A patient's residual limb is generally of a light hue, and, if it is not, it is covered in a thin white sock. A camera is mounted with a lens which ensures that parallel horizontal incoming rays converge at the camera, Figure 3.6. The limb is viewed against a dark background, giving an image such as that shown
in Figure 3.7. A computer scans each line of the picture, identifying at which point the image changes from dark to light and back to dark again. The lens configuration ensures that the image on the computer screen is correct, each line on the screen representing a horizontal scan.

The camera is now swung round the residual limb in a horizontal arc through 180 degrees. This gives a series of images from regularly spaced angles. Restricting our consideration to one horizontal cross-section, the analyses of the images give rise to a number of tangent lines to the object in that cross-section. These tangent lines define a convex area within which the object must lie, Figure 3.8. If the images are taken at closely spaced angles, this area can give a good approximation to the shape of the cross-section of the object. Piecing together the information from all the cross-sections, we know the shape of the whole object. Duncan and Mair have demonstrated a similar method to reproduce a human leg [DUNCAN and MAIR 83].
This edge detection technique works in the above cases because the object is approximately cylindrical. This allows the information from different angles to cover the whole surface, and to be easily pieced together. There is one further constraint on the object which may not be apparent at first sight. The object must be convex in each of its cross sections perpendicular to the axis of camera rotation. A cross section which would not be picked up is shown in Figure 3.9. In the case of residual limb digitisation, this has not been found to be a significant problem because concavities are uncommon, and when they do occur they are usually in limbs which are too severe to be treated with the CASD program anyway.

The form of the data is in slices, but this will be appropriate for a surface model of an object with such an easily defined axis. For the CASD program, the silhouette information for each slice is used to calculate radial values from the central axis at regular intervals, Figure 3.10. This is the form in which the data is stored.
3 Data Capture

Figure 3.9 A cross-section which cannot be picked up by a silhouetting method.

3.4 Ultrasonic and Other Radiation Methods

The final category of methods consists of ultrasonic and other radiation methods such as x-rays, CAT scans and Magnetic Resonance Imaging, (MRI). These methods are particularly appropriate if information other than surface shape is required since they all possess the ability to see features below the skin. However, each of them acts in one plane or slice at any time, and so for coverage of the whole object, a series of slices must be taken. X-rays, CAT scans, Magnetic Resonance Imaging and Ultrasonic scans are discussed here briefly. For a more in depth discussion, the reader is referred to a text dedicated to that purpose, such as [WEBB 88a]. Included there is an article by Webb about how the various medical imaging processes were discovered and introduced [WEBB 88b].

A major factor with the radiation techniques is that they all required subsequent digitisation. The requirement is that an operator must pore over the scans and indicate the points which should be digitised.
Figure 3.10 A silhouetting technique can be used to calculate radial values for data at regular angular intervals.

X-Rays

X-Rays were discovered by Röntgen in 1895 when he was professor of Physics at Wurzburg, and soon x-ray machines were readily available. Today they are widely used to gain an understanding of bone structures. This is because the only tissues which absorb x-rays in large quantities are the bones, and they are therefore the major feature which show up on an x-ray. X-rays are therefore not applicable for prosthetics and orthotics when the surface shape is of particular importance.

CAT Scans

CAT scans are an x-ray based technique, where a series of x-rays are taken parallel to a fixed axis. Therefore, although they suffer from the same restrictions as x-rays, it is possible to build up a three-dimensional picture from CAT scans.
Magnetic Resonance Imaging

Magnetic Resonance, also known as Nuclear Magnetic Resonance, (NMR), was discovered separately in 1946 by two teams who shared a Nobel prize for the discovery [BLOCH et al 46, PURCELL et al 46]. Further detail is given by Leach on how the effects of excitation of protons by the magnetic field are observed in localised areas [LEACH 88]. Magnetic resonance can detect many different types of soft tissue, including skin, and could therefore be used in Prosthetics and Orthotics. Bones are detected by their absence in an image, and a significant advantage is that the scanning process has no known side effects. A drawback is that one image gives only a slice of information, and to obtain a number of slices, the patient is required to remain still for a considerable amount of time.

Ultrasonics

Ultrasonic techniques are the adaption of echo-ranging techniques to the determination of the structure of biological tissues. They were first developed in the engineering industry for non-destructive examination of metallic structures. However, by using ultrasonic waves of a different wavelength, they can be used to give information about human tissue [BOND 82]. The first application to a human part was by Wild and Reid [WILD and REID 52].

3.5 Remarks

As was mentioned in the introduction to this chapter, there are a number of factors which help to determine the data capture method appropriate to a particular situation.

One choice is whether to use a contact or non-contact method. If direct contact with the object being digitised is undesirable because this would lead to shape distortion, then the practicality of taking measurements from plaster casts, or some similar technique, must be assessed. As explained in section 3.2, plaster
casts are usually perfectly acceptable in prosthetics and orthotics, and often are the first stage in the traditional craft-based approach. Indeed an advantage might be that casts can be taken on location and digitised at one central site.

Consideration of an optical method requires thought about the lighting. For those methods where the user picks points from photographs there is no great problem. For Moiré fringes and other patterns to be visible, the ambient light must not be too bright. Lasers of sufficient strength will not be dangerous provided that the source is not looked into directly with the naked eye, but silhouetting will require careful treatment. Duncan and Mair suggest that a bright light is placed behind the object, and others suggest a nylon sock or other covering is worn to counter the lucidity and reflectivity properties of skin [DUNCAN and MAIR 83].

Radiation methods may depend upon whether a scanner is already available, since all the appropriate scanners are very expensive. However, if this implies shared use of the machine, the volume of patients that can be treated in this manner will be limited. Moreover, the scanner may need to be specially set up to cater for the appropriate body portions.

Practical considerations imply that in prosthetics and orthotics where the part being digitised is limited in size and where all the data can be captured in one session, it would seem sensible to be restricted to one technique for data capture. Otherwise difficulties of data fusion may occur [TRAVIS et al 91].

The shape of the object being digitised gives rise to certain factors. If the method is only able to capture information from one view at any moment, then information from a number of views will have to be combined. If this requires movement of the scanner, camera or object, this can lead to significant errors in piecing the view together. Methods which suffer in this regard are the optical methods. However, if the shape is always going to be cylindrical, then this can be overcome by limiting any movement involved to be rotational. This case would depend on whether a general or specific CAD system were required. The methods which can most easily cope with general shape are radiation methods and the hand-held sensors. Mechanically-held sensors are more difficult to
apply because supervisionary software would be required to do the job of ensuring the whole surface is covered. With a hand-held sensor, it is the operator who ensures this coverage.

A further consideration would be the amount of effort required by the operator. While plaster casting for tactile sensing takes time, and some practice is required to master the technique, this can be gained fairly easily and the speed is considered satisfactory in craft methods. Photogrammetry however is very labour-intensive and slow. Radiation methods and Moiré fringes can be similarly intensive if much user-controlled analysis were required. This may lead to rejection of these methods.

In this project it was considered advisable to be able to detect marker points and add them to the digitised data file. Most of the methods discussed in this chapter are able to pick up suitable markers placed on the surface. However, silhouetting might find the markers difficult to identify, and an automatic mechanically-held sensor might miss them altogether.

The form of the data will be a file of \((x,y,z)\) coordinates of points lying on the surface of the object, possibly with some marker points also in the file. Using radiation and silhouetting methods, the data would naturally lie in slices, and with these and other methods requiring user specification of the points to be digitised, there can be an enforced structure of the file. However, it may be easier for some methods, like structured light triangulation and hand-held sensors to yield a random covering of the surface. Therefore, although a CAD system might receive exactly the data it requires for a surface model, it should be able to handle a random covering of the surface to a previously specified density, \(r\), say. That is that there is no circular area on the surface of radius \(r\) which contains no data point in the file. An ideal data capturing technique would be to allow visualisation of the captured points as the digitisation is taking place, thus enabling the user to see that the surface is entirely covered. Methods of editing the data file and extracting the information necessary for the surface model will be discussed in subsequent chapters.
Chapter 4

VIEW 3D - An Editor for Three-Dimensional Data Files

4.1 Introduction

4.2 Data Format

4.3 Three-Dimensional Viewing
   4.3.1 Three Screen Viewing
   4.3.2 Rotations
   4.3.3 Viewing Cross Sections and Visibilities
   4.3.4 Cross Sections

4.4 Three-Dimensional Editing
   4.4.1 A Three-Dimensional Cursor
   4.4.2 Editing Individual Points
   4.4.3 Editing Using the Visibilities

4.5 Remarks
4.1 Introduction

Once data have been captured by some scanner or other digitisation device, as described in Chapter 3, and a file of (x,y,z) coordinates has been produced, there will be a need to check the file. This is first to ensure that the capturing technique functioned properly, and secondly to edit out any erroneous points which may have been captured, as for example can be picked up by an optical scanner at the extremes of its field of view. These purposes necessitate the ability to view, examine in detail and edit a three-dimensional data file, ideally by use of an interactive graphics program on a computer. Furthermore, Chapter 3 suggested that it would be advantageous to be able to locate marker points on the object. Therefore it would be useful if the editing program were able to distinguish between the data and marker points, display them in a distinctive manner, and allow the marker points to be moved or even added at this stage.

This chapter describes the program VIEW3D which has been developed as part of this project. The main features of the program are listed here and expanded in subsequent sections. The program:-

1. Reads in a data file consisting of a stream of (x,y,z) points with marker points optionally included in the file;

2. Provides an overall view of the entire data file in an orthogonal third angle projection with data and marker points clearly distinguishable;

3. Allows the rotation of the file to give a view from any angle;

4. Permits only a portion of the data file to be shown, thus enabling a clearer view of that portion of the file;

5. Shows successive cross-sections of the data file in a user-defined direction;

6. Allows interactive graphical deletion, editing and addition of data points and marker points.
Table 4.1 The Format for a VIEW3D Data File

While there are commercially available packages, such as MacSpin, which have many of these features, the VIEW3D program was developed for three reasons. First, there was not an inexpensive program suitable for the computer available to the project; second, the relevant programs were not able to distinguish between data points and marker points, and to display them in different ways; thirdly, the available programs did not provide the feature of interactively deleting, editing and adding points to the file. In this chapter, the format of the data file for the program and the reasoning behind the choice of features are explained, although it is felt unnecessary to give the actual code in the present work.

4.2 Data Format

This section documents the format of data file required by the editor VIEW3D, and used throughout the current work, indicating how feature (1) of the introduction to the chapter is provided. It is suggested that this format is suitable for a general three-dimensional data file. It was found to be straight-forward to write post-processing modules which converted the data produced by individual scanners of Chapter 3 to the form suggested here.
The file format used is a text (ASCII) file, as shown in Table 4.1. Text format like this is not an efficient use of disk space, but is used for readability of the file. Each line of the file contains the x, y and z coordinates of one point, separated by 'white space' characters and a 'Label' attached to the point. The label is how the program distinguishes between data points and marker points. A label of '1' is attached to data points, and '2' to marker points. However, each label can optionally be omitted, but if it is, the program assumes that the point is an ordinary data point. The reason for this is that it allows a very simple file of (x,y,z) coordinates to be read. Furthermore, it would be a simple extension of the program to allow more than one kind of marker point, if desirable. Then, for example markers indicating bony prominences could be distinguished from markers indicating scar tissue.

4.3 Three-Dimensional Viewing

4.3.1 Three Screen Viewing

To understand a three-dimensional data file, it is not enough to have a simple view of the points in the file from one direction. At the very least there must be some kind of perspective information in the view. When the file is of the form considered in this chapter, namely a large number of points with (x,y,z) coordinates, the information can be given by the size or colour of the dot representing each data point. The size could be largest for the closest points and least for the points furthest from the viewer, the colour could vary from red for the closest points to blue for the furthest points. However, there are two major disadvantages in this approach. First, information can easily be hidden when a point is obscured from view by a nearer point. Second, practice with this approach shows that the information is difficult to interpret, making editing cumbersome.

The approach adopted for VIEW3D is to view the data file in a third-angle projection from three orthogonal views, with the fourth quadrant of the screen reserved for textual information when required. This is similar to the approach adopted by other engineering packages, such as AutoCad and I-DEAS.
Figure 4.1 The manner in which VIEW3D displays a data file comprised of points on the edges of a bevelled cube. Note that in this instance data points on the back of the cube are obscured by points on the front.

[AUTOCAD, IDEAS]. An example of data points taken from a bevelled cube with a marker point on the bevelled face is shown in Figure 4.1. Since each of the three views of the file provides an overall view of the entire file, this is how feature (2) is provided. It should be apparent immediately if the data capturing device has seriously malfunctioned.

4.3.2 Rotations

When physically examining an object, it is natural to turn it round in one’s hand and look at it from different angles. This is because the turning process aids understanding of the three-dimensional shape of the object since the points on the object move in certain ways relative to one another. To enhance the depth perception of the three-dimensional data files, it is advisable to have this ability to rotate the data file and look at it from a different angles within the editor.
Figure 4.2 The bevelled cube data file of the previous figure is rotated about an axis perpendicular to the top left quadrant. The previously obscured data points are now visible.

Feature (3) is therefore included in the program. The rotations can be at any angle, and are screen fixed, but relative to any of the three current views. The effect of a 30° rotation on the cube of Figure 4.1 is shown in Figure 4.2.

An extension to rotations is the ability to animate the data file as it is rotated. This is a feature that can further enhance depth perception, and is included in MacSpin, for example. Although VIEW3D allows the file to be continually rotated, the drawing routines are sufficiently slow that for a large file of 1000 points or more the rotations are not sufficiently quick to appear as an animation.

### 4.3.3 Viewing Cross Sections and Visibilities

Even with the ability to rotate the data file at will, it may be difficult to obtain a full understanding of the file, especially if there are so many points in the file
Visibilities are defined as the maximum and minimum extents of the data file which are currently in view.

that from any angle many data points appear confusingly close even though they are at different depths. Therefore, for improved definition with large data files, feature (4), the ability to view sections of the data, was provided.

The term 'Visibility' was introduced. Six visibilities were defined as the maximum and minimum extents of the data which was currently visible in planes parallel to the three views shown in the quadrants. In Figure 4.3, the Visibilities are shown by dashed lines near the edges of the views. Figure 4.4 gives examples in which two of these Visibilities have been adjusted and are shown by different positions of the dashed lines. Notice that each Visibility appears in two different views, and therefore in each example, the positions of four solid lines are affected. The Visibilities are adjusted interactively, by graphically 'dragging' them to their desired positions.
Figure 4.4 Alteration of the Visibilities affects the portion of the data file currently in view. Each Visibility appears in two screens and affects the view in the third.

4.3.4 Cross Sections

An extension to the ability detailed in the previous section is suggested by the fact that it is often helpful to look at successive cross-sections of an object in the same direction, thereby gaining further insight into the shape of the data file. Feature (5), Cross Sections is therefore incorporated into the program and gives successive cross-sections in one of the three views. The number and thickness of the cross-sections can be controlled, and examples of the cross-sections of a data file taken from a human foot are given in Figure 4.5.
4.4 Three-Dimensional Editing

4.4.1 A Three-Dimensional Cursor

Editing and working in detail with the individual points of a three-dimension data file requires a technique for identifying which of the many points is under consideration. There are common methods which are ideally suited when there are only a small number of points in the file. For example, when choice of a point is required, one point may be highlighted. If this point is not required, the next point in some predetermined order is highlighted, and so on until the desired point is reached. However, with a typical number of points being 2000 or more in one file, such methods are not suitable. A graphical method of identification is required, and for this purpose a three-dimensional cursor was developed.

When it is required, the cursor appears as a small cross in each of the three views. It can be graphically moved around within any of the views, but since each view only shows information relating to two of the coordinate axes, at least two views
Figure 4.6 Movement of the three-dimensional cursor in one quadrant affects its position in one direction only in each of the other quadrants. Here the movement is in the upper left quadrant.

are required to specify a three-dimensional position for the cursor. An example of the movement of the cursor is given in Figure 4.6, where the cursor is moved in the upper-left view.

4.4.2 Editing Individual Points

Individual data points or marker points can be chosen by use of the three-dimensional cursor, although in practice it may also be necessary to restrict the Visibilities in order to ensure that the correct point is identified. It is a natural extension of this ability to edit, delete and add individual data points and marker points, feature (6). Deletion is achieved by indicating the point to be removed with the three-dimensional cursor. Editing of the position of a point
Figure 4.7 The Visibilities can be used to delete all points which are not currently in view in all three quadrants.

requires both the point and its new position to be indicated, and addition of a new point only requires the specification of the new location with the three-dimensional cursor.

4.4.3 Editing Using the Visibilities

While the previous section has shown sufficient flexibility to fulfil the requirements of editing stated in the Introduction to this chapter, in practice a scanner can often pick up objects other than the one under consideration. This is particularly true for optical scanners which can pick up extra points at the extremes of the field of view, for example. To deal with this situation it is sometimes desirable to have the ability to delete whole groups of points at one time as this can save much time and effort.
VIEW3D offers the ability to delete several points together by using the Visibilities. After rotating the data file and restricting the Visibilities so that only desired points are within the Visibilities, all points which lie outside the Visibilities can be deleted from the data file. An example of this is given in Figure 4.7.

4.5 Remarks

This chapter has discussed an editor for three-dimensional data files, VIEW3D, which has been developed as part of the current project. The features of the program have been indicated, and the reasoning behind their adoption discussed.

As a result of the development discussed in this chapter, it is now straight-forward to read in a data file created by one of the scanners described in Chapter 3, to examine the file, and to edit it. After editing, all superfluous points in the data will have been removed, and a good understanding of the shape of the data will have been gained. The data file is then ready for the next stage of the Computer Aided Design process, reducing the data gathered to that required for the surface modelling.
Chapter 5

A Review of Computer Aided Geometric Design and Surface Modelling Techniques

5.1 Introduction

5.2 Curves in two and three dimensions
  5.2.1 Definition of a curve
  5.2.2 Standard polynomial form
  5.2.3 Bezier curves
  5.2.4 Hermite curves
  5.2.5 B-spline curves
  5.2.6 Matrix representation of a curve

5.3 Interpolation with curves
  5.3.1 The interpolation requirements
  5.3.2 Aitkens algorithm
  5.3.3 Hermite curve interpolation
  5.3.4 Bezier curve interpolation
  5.3.5 B-spline curve interpolation
  5.3.6 Continuity

5.4 Surfaces in three dimensions
  5.4.1 Definition of a surface
  5.4.2 Coons patches
  5.4.3 Tensor product surfaces
  5.4.4 Tensor product Bezier patches
5.4.5 Tensor product Hermite patches
5.4.6 Tensor product B-spline surfaces
5.4.7 Matrix representation of a tensor product surface
5.4.8 Bezier triangles

5.5 Interpolation with surfaces in a rectangular situation
5.5.1 The interpolation requirements
5.5.2 Interpolation with Coons patches
5.5.3 Hermite interpolation
5.5.4 Tensor product Bezier interpolation
5.5.5 Tensor product B-spline interpolation
5.5.6 Continuity
5.5.7 Restrictions of the rectangular situation

5.6 Interpolation with surface in a non-rectangular situation
5.6.1 Shephard's methods
5.6.2 Triangular patch methods
5.6.3 Largely rectangular situation Part 1 - Four patch vertices
5.6.4 Largely rectangular situation Part 2 - Four sided patches

5.7 Manipulative ability of surfaces
5.7.1 Large-scale manipulations
5.7.2 Local manipulations
5.7.3 Micro manipulations
5.7.4 Remarks

5.8 Conclusions
5.1 Introduction

In Computer Aided Design, three dimensional surfaces are a common requirement. In the engineering sector many surfaces, particularly those of mechanical parts, can be generated from geometric primitives, extrusions and rotations. Geometric primitives are shapes such as rectangular blocks, cylinders and spheres. This class of primitives can be extended to include conic shapes such as paraboloids, hyperboloids and ellipsoids, and tori. A paraboloid may be used for example in the design of the reflector for a car headlamp. Extensions to the class of primitives may also include the more versatile cyclides, of which tori are a simple example [PRATT 90]. Extrusions and rotations are formed by creating a profile in two dimensions, either by piecing together simple two dimensional primitives such as circular arcs and straight lines or by freehand tracing, and then extruding or rotating this profile as required. The three-dimensional primitives, extrusions and rotations are joined together using Boolean operations to form more complex objects. After the joining has been effected, the sharp corners are often rounded down or filled in to give smooth edges and fillets where required. While such techniques are often sufficient for mechanical engineering purposes, they are not suitable, however, for designing surfaces in general. In particular, the surfaces required in prosthetics and orthotics cannot be formed from primitives, extrusions and rotations because they involve doubly curved surfaces where the shape in one direction is not determined by the shape in any other direction, and so further techniques are required.

Spurred on by the requirements of large manufacturing sectors, such as the car and aircraft industries, where more general doubly-curved surface shapes are necessary, research has been undertaken to investigate other mathematical representations of surfaces. This field of mathematics is known as ‘Surface Modelling’ or ‘Computer Aided Geometric Design’, CAGD for short. The methods of CAGD have been developed with emphasis on the ability to intuitively design surfaces, and with visual tools to enable the design of aesthetically pleasing surfaces. Most surface models are generalisations of simpler curve techniques which were developed first. There is a variety of surface models, reflecting the variety of topologies and shapes required.
However, many methods are restricted in the topologies to which they can be applied - commonly distortions of flat sheets and cylinders are the only topologies available. Some methods are global, meaning that any alteration to the surface will affect the entire surface, and others are local, meaning that a section of the surface can be modified without the rest of the surface being affected. Local surface models are preferable when minor modifications affecting only a small region of the surface are a necessary feature. With global methods, the whole surface is described by one equation or definition. With local methods, there are two types, however. First, those in which the whole surface is determined by one definition, and second, patch solutions, where the surface is comprised of a number of smaller surface sections or patches with appropriate conditions to enforce continuity at the joins of adjacent patches. Where these methods are not general enough to satisfy surface requirements of whole objects, such as the shape of a car bonnet, blending is often required where different sections, designed independently of one another with different surface models, meet.

Typical requirements in prosthetics and orthotics are 'Use some given points to define a surface', 'Manipulate the surface', 'Draw the surface', and 'Carve the surface', and it is the foremost of these which is encountered first. Indeed, although the definition of the surface has implications for the other requirements, it is the formulation of a suitable definition which proves to be the major challenge. The underlying problem with the representation of surfaces is that data points at finite, non-uniform and irregular distances from each other are often thought of as defining a surface. However, a finite number of points does not specify a continuous surface any more than a number of distinct points specifies a continuous curve. A surface model is required to estimate a 'good' surface which interpolates the data. Note that this is a subtly different situation from the engineering industries where the designer is often free to design the shape as he pleases; here the shape of the surface is determined by given data points.

The motivation of this chapter is to review current commonly-used CAGD surface modelling techniques to establish their ability to interpolate data occurring in prosthetics and orthotics. Other reviews of surface modelling with
different emphases are [BARNHILL 83, BÖHM et al 84, CHOI 91, DUNCAN & MAIR 83, PRATT 85, FARIN 90]. Duncan and Mair in particular have a medical interest in their survey. A satisfactory surface model will need to be a local method, and capable of handling the particular topologies encountered. One model should be capable of dealing with the whole body portion under consideration, and blending should be avoided if possible. Since surface methods are often generalisations of curve methods, a review of various curve representations is given in section 5.2, and the application of the methods to interpolation problems is discussed in section 5.3. Similarly, after a review of surface representations in section 5.4, the application of these methods to interpolation problems is discussed, first in a particular case in section 5.5, and then in the general case in section 5.6. The manipulative abilities of the surface models presented are discussed in section 5.7, and section 5.8 concludes the chapter with brief remarks. It will be found that further development of a surface model is required, and this will be the subject of the following chapter. Also, although it may be possible to capture data in a file structure ideal to the final surface model, it may at other times be necessary to extract from the data captured the information required by the surface model. This will also be discussed later in the project.

5.2 Curves in Two and Three Dimensions

Notation

Before specific curves are defined in detail, some remarks are given about the notation used in this chapter and subsequently in the thesis. Points in two and three dimensional space will be in terms of a Cartesian coordinate system. That is, each point will be defined by $x,y,z$ coordinates, or distances in three perpendicular directions from some 'origin' or zero point. A point in space can be represented by a coordinate vector, and vectors will be denoted by lower case letters in bold type. Thus for example
Matrices will be denoted by upper case letters in bold type, for example

\[
M = \begin{pmatrix}
1 & -3 & 3 & -1 \\
0 & 3 & -6 & 3 \\
0 & 0 & 3 & -3 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

\(M^T\) is the transpose of matrix \(M\). In this instance

\[
M^T = \begin{pmatrix}
1 & 0 & 0 & 0 \\
-3 & 3 & 0 & 0 \\
3 & -6 & 3 & 0 \\
-1 & 3 & -3 & 1
\end{pmatrix}
\]

5.2.1 Definition of a Curve

Since there is no reason that curves should necessarily be restricted to two-dimensional space, the definitions here and subsequent development will be in terms of three-dimensional space. There are three possible forms of definition for a curve in three-dimensional Cartesian space - 'Dependent Variable', 'Implicit' and 'Parametric'.

Dependent Variable Form

In this form, two of the Cartesian coordinates are dependent upon the third, giving rise to the name. For example, if \(y, z\) are dependent upon \(x\), then the curve can be written
\[ y = f(x) \]
\[ z = g(x) \]

In this form, the \( y, z \) components can only take on one value for any \( x \) value. The form is, therefore, not suitable for representation of curves which double back on themselves or curves which are closed. A circle is an example which cannot be given in this form.

**Implicit Form**

The implicit form of a curve is defined by two conditions on each point, and can be written

\[ F(x, y, z) = 0 \]
\[ G(x, y, z) = 0 \]

All curves can be represented in this form. For example, a circle centre \((a, b, 0)\), radius \(r\) and contained in the plane \(z = 0\) is given by

\[ F(x, y, z) \equiv (x - a)^2 + (y - b)^2 - r^2 = 0 \]
\[ G(x, y, z) \equiv z = 0 \]

It is often difficult, however, to find a method of calculating points on the curve because this requires either the solution of the equations, or knowledge of some point on the curve, and a technique for finding a neighbouring point on the curve. Both of these can be very tricky, or even impossible, although the example given here is fairly straightforward. Drawing would similarly require determination of a point on the curve and subsequent evaluation of an algorithm for tracking along it, sometimes known as a 'marching method'.
Parametric Form

The third form, Parametric, is also known as the 'Explicit' form because the curve is expressed explicitly in terms of a parameter, $t$, say. The definition is

\[
x = x(t) \\
y = y(t) \\
z = z(t)
\]

and the point $c(t)$ on the curve is given by

\[
c(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}
\]

As the parameter, $t$, takes on all possible values within its allowed range, which is a continuous portion of the Real numbers, the point $c(t)$ will pass through all points on the curve. In practical curves, there is also the restriction that as $t$ varies continuously, the point $c(t)$ will move continuously along the curve. Commonly the range for $t$ is $0 < t < 1$. The circle given above in its implicit form can be written in parametric form, for $0 < t < 1$,

\[
x(t) = a + r \cos(2\pi t) \\
y(t) = b + r \sin(2\pi t) \\
z(t) = 0
\]

However, in parametric form the definition is not unique; for example the same circle, again for $0 \leq t \leq 1$, can be written

\[
x(t) = a + r \sin(2\pi t^2) \\
y(t) = b + r \cos(2\pi t^2) \\
z(t) = 0
\]
In parametric form the curve can be calculated at any point by substituting the appropriate value of $t$, and the curve can be drawn after repeated calculation at increasing values of $t$ covering the entire allowable range. If the range of $t$ is $0 \leq t \leq 1$, then a reasonable sketch is often achieved by using values of $t$ at intervals of 0.1, and much greater accuracy with intervals of 0.01. Because of the ease of calculating points on a curve in parametric form, most curves, and indeed surfaces, in CAGD are expressed in this form.

A common technique in curve surface modelling is Linear Interpolation of two points, and is best described by an example. If $p, q$ are two points in space, then the parametric form of the straight line, $c(t)$, which interpolates to these points is given by

$$c(t) = (1-t)p + tq$$

where $t$ is restricted to $0 \leq t \leq 1$. Then $c(0) = p, c(1) = q$, and when $0 \leq t \leq 1$, $c(t)$ lies on the straight line segment between the points $p, q$. Linear interpolation has already been encountered in the UCL CASD system reducing helical data to sliced data in section 2.5, for example.

5.2.2 Standard Polynomial Form

The standard polynomial, or monomial, form of a curve $c(t)$ of degree $n$ is

$$c(t) = a_0 + a_1 t + a_2 t^2 + \ldots + a_n t^n$$

In this form, the polynomial can be quickly and stably evaluated, but the points (or coefficients) mentioned in the equation of definition have no geometrical relationship to the shape of the curve, except at the point corresponding to $t = 0$, where the curve passes through point $a_0$ with tangent direction $a_1$. Moreover, for a standard polynomial curve to interpolate $n$ points, a curve of degree $n$ is required, and practice shows that these curves are oscillatory, giving rise to 'unnatural' shapes.
5.2.3 Bezier Curves

Bezier curves are defined and discussed in this section - the definition is equation (5.6) - but the definition has as its coefficients Bernstein polynomials, and so these are first introduced and their major features listed. The Bernstein polynomials of degree \( n \), which are not vectors but scalars (or numbers), are defined by

\[
B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}
\]  

(5.1)

where \( \binom{n}{i} \) is the binomial coefficient, and \( 0 \leq i \leq n \).

The Bernstein polynomials have several important properties:

**Partition of Unity** The sum of all the Bernstein polynomials of degree \( n \) is equal to 1 for all values of \( t \),

\[
\sum_{i=0}^{n} B_i^n(t) = 1
\]  

(5.2)

**Positivity** For \( 0 \leq t \leq 1 \), the Bernstein polynomials are non-negative,

\[
B_i^n(t) \geq 0 \quad \text{for } 0 \leq t \leq 1
\]  

(5.3)

**Symmetry**

\[
B_i^n(t) = B_{n-i}^n(1-t)
\]  

(5.4)
Recursion The Bernstein polynomials of degree $n$ are a linear combination of the Bernstein polynomials of degree $n - 1$,

$$B_n^s(t) = (1-t)B_{n-1}^{s-1}(t) + tB_{n-1}^{s-1}(t) \quad (5.5)$$

The first of these three properties requires the Binomial theorem for its proof, the others follow from simple algebraic manipulation. Figure 5.1 shows the Bernstein polynomials in the case $n = 4$.

At this stage, Bezier curves can be defined. The Bezier curve of degree $n$ over the $n + 1$ points $b_0, ..., b_n$ is defined by

$$b(t) = \sum_{j=0}^{n} b_j B_j^s(t) \quad (5.6)$$

The points $b_0, ..., b_n$ are the 'control points' or the 'Bezier points' of the curve, and the curve is of 'degree' $n$. The polygon constructed by connecting successive
control points by straight line segments can be referred to as the 'control polygon' or the 'Bezier polygon'. Examples are given in Figure 5.2. The definition of a Bezier curve yields several properties:-

**Intuitive 'Shape'** The control points are so called because the shape of a Bezier curve roughly resembles the layout of its defining points, see Figure 5.2. Furthermore, if only one of the points is altered, then the shape of the curve will be most affected in the region of that point, although the shape of the entire curve is altered. This property is especially of interest in designing aesthetic shapes, and has applications in prosthetic and orthotic surfaces, discussed later.

**Affine Invariance** An Affine transformation is a transformation of space which preserves barycentric combinations (weighted averages) of points. In particular, rotations, translations, reflections and scalings are all affine transformations. The Partition of Unity property of Bernstein polynomials (5.2) implies that if the points which determine the Bezier curve are subjected to an affine transformation then the curve will follow the same affine transformation.
Convex Hull Property. The convex hull of a point set \( \{ b_i \} \) is the intersection of all convex sets containing the \( \{ b_i \} \), and the positivity of the Bernstein polynomials implies that a Bezier curve will always lie within the convex hull of the points defining the curve, Figure 5.3. In particular if all the defining points lie in a straight line, then the Bezier curve will be that straight line reproduced.

Endpoint Interpolation. From the values of the Bernstein polynomials at \( t = 0 \) and \( t = 1 \), the Bezier curve starts at point \( b_0 \) when \( t = 0 \) and ends at point \( b_n \) when \( t = 1 \).

Symmetry. From the symmetry of the Bernstein polynomials, the points \( b_0, \ldots, b_n \) and the points \( b_n, \ldots, b_0 \) yield the same Bezier curve, but traversed in the opposite direction as \( t \) increases. Mathematically expressed, the property is

\[
\sum_{j=0}^{n} b_j B_j^n(t) = \sum_{j=0}^{n} b_{n-j} B_j^n(1-t) 
\]  

(5.7)
Invariance Under Barycentric Combinations The process of forming the Bezier curve from the Bezier polygon leaves barycentric combinations invariant: for $\alpha + \beta = 1$ and two Bezier curves, defined by $b_0, \ldots, b_n$ and $c_0, \ldots, c_n$, this gives

$$\sum_{j=0}^{n}(\alpha b_j + \beta c_j)B^*_j(t) = \alpha \sum_{j=0}^{n} b_jB_j^*(t) + \beta \sum_{j=0}^{n} c_jB_j^*(t) \quad (5.8)$$

This means that it makes no difference whether a weighted average of control points is taken to define the Bezier curve, or a weighted average of two Bezier curves of equal order is taken. This property will be used later to justify the establishment of Bezier surfaces.

The de Casteljau Algorithm The Recursion property of Bernstein polynomials implies that a Bezier curve can be obtained from the control points by repeated linear interpolation. This repeated interpolation is given the name 'The de Casteljau Algorithm' since an independent development of Bezier curves was constructed by de Casteljau using this technique [DECASTELJAU 63].
The algorithm reads:

\[
\begin{align*}
\mathbf{b}_j^0(t) &= b_j, && 0 \leq j \leq n \\
\mathbf{b}_j^k(t) &= (1-t)\mathbf{b}_{j+1}^{k-1} + t\mathbf{b}_j^{k-1}(t), && 1 \leq k \leq j \leq n
\end{align*}
\] (5.9)

The \(n\)th step of the algorithm gives the appropriate point on the Bezier curve,

\[
\mathbf{b}_n^*(t) = \mathbf{b}(t)
\] (5.10)

This is shown in a diagram for the case \(n = 4\) in Figure 5.4. To prove the result requires induction on the expression in (5.11). However, the proof is not required as part of this review, and for its working the reader is referred to [FARIN 90]. For small order, this algorithm is a stable and efficient method of calculation, although for larger \(n\) the efficiency decreases.

\[
\mathbf{b}_j'(t) = \sum_{s=0}^{\infty} b_{j+s} B_s'(t)
\] (5.11)

**Derivative of a Bezier curve** The derivative of a Bezier curve can be calculated and written in terms of Bernstein polynomials:

\[
\frac{d}{dt} (\mathbf{b}(t)) = \sum_{j=0}^{\infty} b_j \frac{d}{dt} (B_j^*(t))
\]

\[
= n \sum_{j=1}^{\infty} (b_j - b_{j-1}) B_j^{*-1}(t)
\] (5.12)

In particular note that the derivative at \(t = 0\) is

\[
\frac{d}{dt} (\mathbf{b}(0)) = n \sum_{j=1}^{\infty} (b_j - b_{j-1}) B_j^{*-1}(0)
\]

\[
= n(b_1 - b_0)
\] (5.13)

In words, the tangent at the start of the curve is in the direction of the second control point. This can be seen from the examples in Figure 5.2.
If the difference operator is defined as $\Delta^r b_j = \Delta^{r-1} b_{j+1} - \Delta^{r-1} b_j$ with $\Delta^0 b_j = b_j$, then by repeated application of the derivative equation,

$$\frac{d^r}{dt^r} b(t) = \frac{n!}{(n-r)!} \sum_{j=0}^{n-r} \Delta^r b_j \binom{n}{j} (1 - t)^{n-j}$$

In particular, at $t = 0$,

$$\frac{d^r}{dt^r} b(0) = \frac{n!}{(n-r)!} \Delta^r b_0$$

Thus the $r$th derivative of a Bezier curve at its endpoint depends only on the $r + 1$ control points nearest that endpoint.

Avoidance of Looped Bezier Curves An important property which can be gleaned from (5.12) is that if the control points are monotonic in one direction, then the curve cannot double back on itself in that direction, and must be both cuspless and loopless.

Degree Elevation of a Bezier Curve A Bezier curve of degree $n$ can be expressed as a Bezier curve of degree $n + 1$. This has the effect of increasing the flexibility of a Bezier curve since more control points imply greater flexibility. The process is known as degree elevation, and can be achieved if there exist points $b^*_0, b^*_1, \ldots, b^*_n$ which satisfy the condition

$$\sum_{j=0}^{n} b_j \binom{n}{j} t^{n-j} (1 - t)^{-j} = \sum_{j=0}^{n+1} b^*_j \binom{n+1}{j} t^j (1 + t)^{n+1-j}$$

Multiplying the left hand side by $(t + (1 - t))$,

$$\sum_{j=0}^{n} b_j \binom{n}{j} [t^j (1 - t)^{n+1-j} + t^{j+1}(1 - t)^{n-j}] = \sum_{j=0}^{n+1} b^*_j \binom{n+1}{j} t^j (1 + t)^{n+1-j}$$
Comparing coefficients of $t^i(1-t)^{n+1-j}$,

$$b_j^{n+1} = b_j^n + b_{j-1}^n$$

or, rearranging,

$$b_j^* = \left(1 - \frac{j}{n+1}\right)b_j + \left(\frac{j}{n+1}\right)b_{j-1}$$

(5.15) is a formula for using linear interpolation to find the control vertices of a Bezier curve of increased order. Degree elevation will be used in this project to increase the flexibility of Bezier curves and surfaces without affecting their continuity conditions, and is demonstrated in Figure 5.5.
Variation Diminishing. Repeated application of degree elevation can yield another impressive property of Bezier curves, namely that a Bezier curve will have no more intersections with a plane than its control polygon does [FARIN 90]. This property is known as 'Variation Diminishing'. Moreover, if the control polygon turns through less than 180°, then the curve turns through less than 180° [LAU 88]. In practice, variation diminishing imposes a pleasing aesthetic shape upon the curve.

Subdivision of a Bezier curve. A Bezier curve can be subdivided at any point into two smaller Bezier curves of the same order. To subdivide a Bezier curve $b(t)$ at a point $b(c)$, it turns out that each of the portions of the curve $b(0)$ to $b(c)$ and $b(c)$ to $b(1)$ are each Bezier curves, with control points deduced from the de Casteljau algorithm for the point $b(c)$, Figure 5.6. Since the proof is not...
particularly easy to follow for someone with a non-mathematical background, and since the proof is not an integral part of this text, the reader is referred to the two elegant proofs given in [FARIN 90].

**Forward Difference Calculation** One further property of Bezier curves is the forward difference calculation property. It is not exclusive to Bezier curves, indeed it can be used for any polynomial function, but it will be used for the calculation of Bezier curves and surfaces during the project, and this is a convenient point to derive it.

Define

\[ \Delta b(t) = b(t + h) - b(t) \]
\[ \Delta^k b(t) = \Delta^{k-1} b(t + h) - \Delta^{k-1} b(t) \]

Now if \( b(t) \) is a polynomial of order \( n \) then \( \Delta b(t) \) is a polynomial of order \( n - 1 \), and so on for higher order differences. In particular \( \Delta^n b(t) \) is constant. This suggests the following table

<table>
<thead>
<tr>
<th>( \Delta^n b(t) )</th>
<th>( \Delta^{n-1} b(t) )</th>
<th>( \Delta^{n+1} b(t) )</th>
<th>( b(t + h) )</th>
<th>( b(t) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta^n b(t + h) )</td>
<td>( \Delta^{n-1} b(t + h) )</td>
<td>( \Delta^{n+1} b(t + h) )</td>
<td>( \Delta^n b(t) )</td>
<td>( b(t) )</td>
</tr>
<tr>
<td>( \Delta^n b(t + 2h) )</td>
<td>( \Delta^{n-1} b(t + 2h) )</td>
<td>( \Delta^{n+1} b(t + 2h) )</td>
<td>( \Delta^n b(t + 2h) )</td>
<td>( b(t + 2h) )</td>
</tr>
</tbody>
</table>

The first row can be calculated at the initial point. The left hand column is constant and, therefore, subsequent rows can be calculated left to right using the identity

\[ \Delta^k b(t) = \Delta^{k+1} b(t - h) + \Delta^k b(t - h) \] (5.16)

The right hand column is the sequence of points on the curve at equal parameter intervals. Note that this calculation is very efficient since after the establishment of the first row, no multiplications are required, but just additions. Moreover, only a one-dimensional array is required for storage of intermediate results.
However, because this technique uses previous results to determine subsequent results, any errors are cumulative. The errors are controlled by the accuracy of the computer. For a small number of iterations the result is good; as the number of iterations becomes very large, the result becomes progressively worse.

5.2.4 Hermite Curves

The Hermite cubic curve is commonly found in the literature [eg CONTE & DEBOOR 80, SCHUMAKER 81]. Although closer examination of the definition of a Bezier curve and its derivative shows that the cubic Hermite curve is a Bezier curve of degree three, it is presented here in a separate section because of the frequency with which it occurs in the literature. A Hermite cubic curve is defined as a curve, $h(t)$, which is a function of the interval $[0, 1]$, and which passes through points $a$ at $t = 0$ and $b$ at $t = 1$, and has derivatives $s$ and $t$ at those points respectively, Figure 5.7. The equation of the curve, $h(t)$, is given by
\[ h(t) = (1 - t)^2 (1 + 2t)a + (1 - t)^2 ts + (3 - 2t)t^2 b - t^2 (1 - 2t)t \] (5.17)

This is in fact a cubic Bezier curve with

\[
\begin{align*}
    b_0 &= a \\
    b_1 &= a + \left( \frac{1}{n} \right) s \\
    b_2 &= b - \left( \frac{1}{n} \right) t \\
    b_3 &= b
\end{align*}
\] (5.18)

Similarly, a quintic Hermite curve, which in addition to the endpoint and tangent interpolation of the cubic Hermite curve also interpolates curvature values at its ends, is a Bezier curve of degree five.

### 5.2.5 B-Spline Curves

B-spline curves are a further type of polynomial curve and, when extended to surfaces, are probably the representation found most frequently in practice. They were first developed as part of the development of parametric spline curves at Boeing in the early 1960's - spline curves being piecewise polynomial curves with certain continuity restraints. B-splines were first considered in detail by de Boor [DEBOOR 72], but the analysis of the B-splines was mostly concerned with mathematical properties. Gordon and Riesenfeld were the first to amalgamate the theory of B-splines and the philosophy of Bezier curves and exploit their use for Computer Aided Design [GORDON & RIESENFELD 74]. Since then, many of their properties have been examined in great detail, and the major results relevant to the current study are presented in this section.

B-spline curves are partly determined by a number of 'control points' which, in a similar manner to the control points of a Bezier curve, are in a certain order and affect the shape of the curve. However, for the full definition of a B-spline curve, a sequence of non-decreasing real values, known as a 'knot sequence', is
also required. The knot sequence governs how the relevant point on the curve is calculated from the positions of the control points. B-spline curves can have as many degrees of continuity as desired, although curvature continuity is in practice often enough. If consecutive values in the knot sequence are equal, the curve loses a degree of continuity at that point. This means that B-spline curves can be constructed which contain sharp corners, although they are smooth at every other point. Furthermore, a point on a B-spline curve is only dependent upon the control points nearest to that point. After some calculation it follows that by setting a number of consecutive control points on a straight line, a B-spline curve can contain a straight line segment. The relative intervals between the successive knots control how closely the B-spline curve is to each of the control points at any point - the knots can be set up so that the curve hugs an individual control point much more tightly than the other points.

In fact, a B-spline curve with \( r \) control points turns out to be a curve comprised of \( r - 1 \) Bezier curves joined end to end with necessary conditions on the Bezier curve control points to ensure continuity between the adjacent Bezier curves.

Recall that for the definition of Bezier curves, Bernstein polynomials were required. Similarly for B-spline curves, basis functions are again required. These are known as 'B-splines' (as opposed to 'B-spline curves') since they are actually splines [see SCHUMAKER 81], although to avoid confusion, they will be referred to as 'B-basis functions' in this section. Define the B-basis functions over the knot sequence \( \ldots t_{i-1} \leq t_i \leq t_{i+1} \ldots \) by the following formulae, the recursive definition being named the Cox - de Boor algorithm after its discoverers,

\[
N_i^0(t) =  \begin{cases} 
1 & t_i < t \leq t_{i+1} \\
0 & \text{otherwise} 
\end{cases} \\
N_i^k(t) = \left( \frac{t - t_i}{t_{i+k} - t_i} \right) N_i^{k-1}(t) + \left( \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} \right) N_{i+1}^{k-1}(t)
\]

The B-basis functions have the following properties:-
Partition of Unity The sum of the B-basis functions of order k is 1, for any value of t,

\[ \sum_{i=0}^{k} N_i^k(t) = 1 \]  

(5.19)

Positivity The B-basis functions are all non-negative,

\[ N_i^k(t) \geq 0 \quad \text{for all } k, i, t \]  

(5.20)

Local Support There is only a limited range for which any B-basis function is not zero,

\[ N_i^k(t) > 0 \quad t_{i-k} < t < t_i \]

\[ = 0 \quad \text{otherwise} \]
Continuity The $k$th order B-basis functions $N_i^k(t)$, have continuity order $k$.

Now a B-spline curve can be defined. For control points $b_i$, sometimes known as 'de Boor points' and which form the 'control polygon', over the knot sequence \( \ldots t_{i-1} \leq t_i \leq t_{i+1} \ldots \) and B-basis functions already defined, a B-spline curve of 'order' $k$ or degree $k$ is defined as, (compare (5.6)),

\[
b(t) = \sum_{j=i}^{i+k-1} N_j^k(t)b_j \tag{5.22}\]

Examples of example B-spline curves are given in Figure 5.8. B-spline curves have the following properties:

Convex Hull Property From the Partition of Unity (5.19) and Positivity (5.20) of the basis functions, the Convex Hull property follows. From the Local Support (5.21) of the basis functions, the Convex Hull property can be made more severe;
the curve not only lies within the convex area contained within all of the control points, but each section of the curve lies within the convex area contained within the $k + 1$ control points relevant to that section of the curve, Figure 5.9.

**Variation Diminishing** The variation diminishing property which Bezier curves possessed also holds for B-spline curves, as shown by Lane and Riesenfeld [LANE & RIESENFELD 83].

**Differentiability** From the continuity of the basis functions, a B-spline curve of degree $k$ is $k$ times differentiable with respect to its parameter except when $r$ knots are identical, when the curve is $k - r + 1$ times differentiable.

**Intuitive Shape** From the local support property of the basis functions, at any point, a B-spline curve of order $k$ is only dependent on the $k + 1$ control points close to that point, as shown in Figure 5.10. Therefore if a control point is moved, only the portion of the curve near to that control point will be affected.
The de Boor Algorithm Similar to the de Casteljau algorithm for Bezier curves (5.9), B-spline curves can be generated by repeated linear interpolation using the de Boor algorithm,

$$b_i^0 = b_i$$

$$b_i^j = \left( \frac{t - t_i}{t_{i+k-j} - t_i} \right) b_i^{j-1} + \left( \frac{t_{i+k-j} - t}{t_{i+k-j} - t_i} \right) b_{i+1}^{j-1} \text{ for } j = 1, \ldots, k - 1$$

(5.23)

Then for $t_i < t \leq t_{i+1}$, as shown in Figure 5.11, we have

$$b_i^k = b(t)$$

(5.24)

Knot Insertion Note that it is possible to have up to $k - 1$ knots at the same value for the recursion formula to remain valid. To increase the flexibility of Bezier curves, the order was raised, thereby increasing $n$ control points to $n + 1$ control points. This process could be viewed as the first stage of the de Casteljau
B-Spline Curves are Composite Bezier curves

We are now in a position to show this result which will justify our restriction to consideration of Bezier curves. Consider knots $t_i$ and $t_{i+1}$ each repeated $k - 1$ times, referred to as 'multiplicity $k - 1$'. Then consideration of the curve between these limits shows that it is just
the Bezier curve with the de Boor algorithm reducing to the de Casteljau algorithm, Figure 5.13. This yields a two-stage procedure for obtaining the Bezier points for a B-spline curve, namely:-

1. Insert new knots so that all knots have multiplicity $k - 1$;
2. Read off the Bezier control points of each section of the curve. Note that since each insertion of a knot value increases the number of control points by one, the control points at this stage will actually be the Bezier control points.

For further discussion of this result, the reader is referred to [BÖHM et al 84]. Note also, that in Farin’s derivation of a B-spline curve the property is immediate since he defines a B-spline curve as Bezier curves with the necessary continuity conditions [FARIN 90].

**Uniform Cubic B-Spline curves** If the difference between successive knots is a constant, and this is often the case used by designers, then a B-spline curve is
known as 'uniform'. The most common way to achieve this is to set

\[ t_i = i \quad \text{for all } i \]

If the uniform case is used, then the defining equations, and hence the implementation routines, become much simpler than the recursive scheme suggested above. This is because the intervals between successive knots are uniform, and cancel out from the formulae. In the cubic case, which guarantees tangent and curvature continuity, this gives, where \( \theta = t_i - i - 3 \):

\[
N_i^3(t) = \frac{1}{6} (1 - 3\theta + 3\theta^2 - \theta^3)
\]

\[
N_{i+1}^3(t) = \frac{1}{6} (4 - 6\theta^2 + 3\theta^3)
\]

\[
N_{i+2}^3(t) = \frac{1}{6} (1 + 3\theta + 3\theta^2 - 3\theta^3)
\]

\[
N_{i+3}^3(t) = \frac{1}{6} (\theta^3)
\]

**Discussion of B-Spline Curves** It has been shown that B-spline curves are comprised of a number of Bezier curves pieced together with continuity at their joins. The B-spline curve representation has certain advantages over the Bezier form: the total amount of information required for a B-spline curve is less than the total required for the constituent Bezier curves; a B-spline curve of order \( k \) can have as many control points as desired; continuity between the various sections of the curve is guaranteed. However, there is a major disadvantage of B-spline curves for interpolation purposes - the B-spline curve does not interpolate the control points. As will be discussed under curve interpolation, this can be corrected, but at the expense of the result that the curve is no longer 'local' - that is that the positions of all the control points affect the shape of the entire curve, and movement of one point would affect the shape of the entire curve.
Calculation methods for B-spline curves are many and varied. From the forms of the equations, however, it is apparent that calculation of uniform B-spline curves are much more straightforward than non-uniform B-spline curves. Forward differencing methods, similar to those outlined for Bezier curves, can be developed for B-spline curves, and other efficient methods are available for their calculation.

Barsky has generalised the B-spline curve to the Beta-spline curve by including a tension parameter [BARSKY 81, BARSKY 85]. The tension affects how closely the curve hugs the defining control polygon - the higher the tension, the closer the hugging. When the tension has value zero, the curve is a B-spline curve, and when the tension becomes very high the curve becomes the defining control polygon.

It is widely agreed, however, that B-spline curves are a good general solution to challenges requiring curve solutions [BÖHM et al 84, FARIN 90]. B-spline methods are not sufficient to this project only because they suffer from limitations in their applications when the curve methods are extended to B-spline surface methods.

5.2.6 Matrix Representation of a curve

Any polynomial representation for a curve, of which Bezier, Hermite and B-spline curves are examples, can also be expressed in matrix form in terms of the basis \( t^0, t^1, t^2, t^3, \ldots, t^n \). The matrix expression takes the form,

\[
c(t) = (c_0 \ldots c_n) M \begin{pmatrix} t^0 \\ \vdots \\ t^n \end{pmatrix}
\]

For standard polynomial form, as discussed in section 5.2.2, \((c_0 \ldots c_n) = (a_0 \ldots a_n)\) and \(M\) is the identity matrix, giving in the cubic case,
For a Bezier curve, the matrix \( M \) will depend only upon the order of the curve. For a cubic Bezier curve, the equation is

\[
\mathbf{b}(t) = (b_0, b_1, b_2, b_3) \begin{pmatrix}
1 & -3 & 3 & -1 \\
0 & 3 & -6 & 3 \\
0 & 0 & 3 & -3 \\
0 & 0 & 0 & 1
\end{pmatrix} \begin{pmatrix}
1 \\
t \\
t^2 \\
t^3
\end{pmatrix}
\]

Similarly, for the uniform cubic B-spline curve, with \( \theta = t - i - 3 \),

\[
\mathbf{b}(t) = (b_i, b_{i+1}, b_{i+2}, b_{i+3}) \frac{1}{6} \begin{pmatrix}
1 & -3 & 3 & -1 \\
4 & 0 & -6 & 3 \\
1 & 3 & 3 & -3 \\
0 & 0 & 0 & 1
\end{pmatrix} \begin{pmatrix}
1 \\
\theta \\
\theta^2 \\
\theta^3
\end{pmatrix}
\]

This form is much simpler than the recursively defined form above, but it is only for the uniform cubic case. For the cubic Hermite case, the matrix equation is

\[
\mathbf{h}(t) = (a, b, s, t) \begin{pmatrix}
1 & 0 & -3 & 2 \\
0 & 0 & 3 & -2 \\
0 & 1 & -2 & 1 \\
0 & 0 & -1 & 1
\end{pmatrix} \begin{pmatrix}
1 \\
t \\
t^2 \\
t^3
\end{pmatrix}
\]

Matrix form can sometimes be used to give efficient calculation, particularly if matrix multiplication is hard-wired. However, matrix manipulation with curves in this manner can be numerically unstable because of the large number of multiplications required.
5.3 Interpolation with Curves

Having discussed the main types of curves available for modelling purposes in the previous section, the interpolation schemes which use these curves are now discussed. First, though, the interpolation requirements are quantified more precisely.

5.3.1 The Interpolation Requirements

A typical situation for curve interpolation is to find a curve which passes through points $p_i \ 0 \leq i \leq n$ in the order signified by their numbering. At this stage, no restrictions are imposed upon the relative distances between the points, although some methods will give more aesthetically pleasing shapes if the distances between consecutive points are fairly similar. Many of the methods previously mentioned connect only two consecutive points, and so many pieces will have to be joined together to form a curve through all the points. Others will be able to form a curve through all the points with only one definition.

In section 5.2, the curves which were available for designing shape were in parametric form. For these definitions to be applicable to the current interpolation problem with several points, the whole curve should be definable over one parameter, and there will need to be a value $t_i$ for each point $p_i$ such that the curve passes through the point at the specified parameter value. Often these parameter values are not known, and so methods are given for their estimation.

Some of the curves which have been suggested will require the tangent vectors $t_i$ at each of the points $p_i$. Methods for estimating these tangent values are given for those cases in which such estimation is necessary.

After the parameter and tangent estimation methods, Aitken's algorithm, a technique with a simple philosophy for establishing a curve through a number of points, is developed, and then the methods of section 5.2 are applied to the interpolation situation.
Figure 5.14 A common estimate for tangent direction at point $p_i$ is parallel to the line $p_{i-1}p_{i+1}$.

Parameter Value Estimation

Parameter values $t_i$ for each of the points $p_i$ are often not known, but are required by the curve representation used. Estimation methods are, therefore, required. The simplest method to estimate values for the parameter is to set $t_i = i, \quad 0 \leq i \leq n$. However, if the points are not fairly evenly spaced, this can lead to strange results such as undesirable variations in the curvature. A second, commonly used, technique for parameter estimation is to set

$$t_0 = 0$$

$$t_{i+1} = t_i + \|p_{i+1} - p_i\|.$$  \hspace{1cm} (5.29)

Tangent Estimation

It may be that tangent vectors or directions, $t_i$, at the points $p_i$ are known, but if
not they may have to be estimated. A common method for this is to set the
tangent direction at point $p_i$ by the following formula as suggested in Figure
5.14

$$t_i = \text{unit}(p_{i+1} - p_{i-1})$$  \hspace{1cm} (5.30)

where \textit{unit} denotes the action of taking the unit vector in the given direction, ie

$$\text{unit}(n) = \frac{n}{\|n\|}$$

This does not work at the end points, $p_0$ and $p_n$. At these points, a different
approach is required. One solution is to take the Bessel tangents at these points,
described next.

The Bessel tangent at point $p_i$ is obtained by passing a quadratic curve through
$p_{i-1}, p_i, p_{i+1}$ and evaluating the derivative with respect to $t$ at $t_i$. In terms of the $p_i$
the result is

$$t_i^* = \frac{(t_{i+1} - t_i)}{(t_{i+1} - t_{i-1})(t_i - t_{i-1})}(p_i - p_{i-1}) + \frac{(t_i - t_{i-1})}{(t_{i+1} - t_{i-1})(t_{i+1} - t_i)}(p_{i+1} - p_i)$$

$$t_i = \text{unit}(t_i^*)$$  \hspace{1cm} (5.31)

For the end cases, the result is

$$t_0^* = \frac{2}{t_1 - t_0}(p_1 - p_0) - t_1$$

$$t_n^* = \frac{2}{t_n - t_{n-1}}(p_n - p_{n-1}) - t_{n-1}$$

$$t_0 = \text{unit}(t_0^*)$$  \hspace{1cm} (5.32)

$$t_n = \text{unit}(t_n^*)$$

In this case, since $t_i$ is used in the definition of $t_0$, it is important that $t_0$ is calculated
from \( t \) and not \( t' \). A similar condition holds for \( t' \).

A third possibility of tangent direction at point \( p \), is the tangent at \( p \) to the circle through \( p_{i-1}, p_i, p_{i+1} \). The derivation of this direction is left until a simple proof is available in section 6.4.2. For now, we state that this direction is

\[
\mathbf{t}_i = \frac{(p_{i+1} - p_i) - (p_{i+1} - p_i) \cdot \mathbf{n}_i}{\mathbf{n}_i} \quad (5.33)
\]

There is one further case for consideration. At a point \( p \), the tangent plane only may be known, with unit normal \( \mathbf{n}_i \). Call this plane \( \Pi \). Then if the next interpolation point of the curve is \( p_{i+1} \), the tangent direction can be estimated to be in the direction of the projection of \( p_{i+1} \) into the plane \( \Pi \). Mathematically this is

\[
\mathbf{t}_i = \text{unit}((p_{i+1} - p_i) - (p_{i+1} - p_i) \cdot \mathbf{n}_i) \cdot \mathbf{n}_i \quad (5.34)
\]

where \( \cdot \) denotes the vector dot product.

With the exception of the Bessel tangents, the methods of tangent estimation have given the tangent direction, but not its magnitude. This may also be required. Suppose that at point \( p \), the unit tangent in the direction of the curve toward \( p_{i+1} \) is \( t \). Then the commonest method for estimation of the tangent length, \( \alpha \), is

\[
\alpha = (b_{i+1} - b_i) \cdot t_i \quad (5.35)
\]

5.3.2 Aitkens Algorithm

The Aitkens algorithm uses repeated linear interpolation to pass a curve through points \( p \). In the cubic case, the final cubic curve through \( p_0, p_1, p_2, p_3 \) is a linear combination of the quadratic curves through \( p_0, p_1, p_2 \) and \( p_1, p_2, p_3 \)
An interpolating smooth curve cannot possess the variation diminishing or convex hull property since there must be a point $p_i$ and a plane through that point which has all other points on one side of it, and cuts the curve more often than it cuts the defining polygon.

respectively. Similarly, the quadratic curve through $p_0, p_1, p_2$ is a linear combination of the linear curves (straight lines) through $p_0, p_1$ and $p_1, p_2$ respectively. The general form of the Aitkens algorithm is to set $p_i = p_i^0$, and then

$$p_r^i(t) = \left( \frac{t_{i+r} - t}{t_{i+r} - t_i} \right) p_i^{r-1}(t) + \left( \frac{t - t_i}{t_{i+r} - t_i} \right) p_{i+1}^{r-1}(t) \quad r = 1, \ldots, n \quad i = 0, \ldots, n - r \quad (5.36)$$

Then $p_0^0(t)$ is the general point on the interpolating polynomial curve of degree $n$. Two properties are worth noting from Aitkens algorithm. First, the resulting curve does not lie within the convex hull of the original points $p_i$, and second, the curve does not have the variation diminishing property. However, we now show that no smooth (tangent continuous) curve interpolation scheme has either of these properties.
The Variation Diminishing and Convex Hull Properties

Interpolating curves do not possess the Variation Diminishing and Convex Hull properties: Suppose that the points $p_1, \ldots, p_{n-1}$ do not lie on the straight line segment between $p_0$ and $p_n$. Then there is a point $p_i, i \neq 0, n$ and a plane $\Pi$ containing $p_i$ with all other points on one side of the plane. Suppose there is a curve which smoothly interpolates the $p_i$ with tangent $n$ at $p_i$, Figure 5.15. We can assume without loss of generality that $n \in \Pi$. (If it were, then a slightly different plane through $p_i$ could be chosen.)

If the variation diminishing property holds there must be a small $\epsilon$ such that the two points $p_i + \epsilon n$ and $p_i - \epsilon n$ lie on the same side of the plane, otherwise the curve would cross $\Pi$ more times than the original polygon. This is not possible.

If the convex hull property holds there must be a small $\eta$ such that the two points $p_i + \eta n$ and $p_i - \eta n$ lie in the convex hull. These points must lie on either side of the plane $\Pi$. Since all the points $p_j, j \neq i$ lie on one side of $\Pi$, so must the convex hull with the exception of point $p_i$. A contradiction.

Thus any smooth interpolation method will not have either the convex hull or the variation diminishing property over the points to be interpolated.

The CASD High Resolution Method

Recall from section 2.7 that the CASD program uses an estimation method to calculate intermediate points to the known radial values. The method uses the Aitken algorithm on each set of four consecutive radial values distributed at uniformly spaced $t$ values, and interpolates a mid-range value. For example, the values $r_{-3}, r_{-1}, r_1, r_3$ are used at parameter values $t = -3, -1, 1, 3$ respectively to interpolate at $t = 0$. Note, however, that it is not the Aitkens algorithm used over all points $r_i$, but only four consecutive values at any one instance. A diagram of two successive calculated points is shown in Figure 5.16. The previous result
that Variation Diminishing does not carry over to surfaces can be seen from the exceptional cases where the CASD High Resolution Method has to be refined, Figure 2.17.

5.3.3 Hermite Curve Interpolation

Hermite cubic curves can be used to interpolate to the points \( p_i \). If two successive curves are denoted by \( a_1, b_1, s_1, t_1 \) and \( a_2, b_2, s_2, t_2 \), then for the curve to be continuous and continuous in tangent, Figure 5.17, the following conditions are required:-
For two Hermite cubic curves to join smoothly they must have a common end point and parallel tangents at that point.

\[ a_2 = b_1 \]  
\[ s_2 = \alpha t_1 \quad \alpha > 0 \]

If \( \alpha < 0 \) the curve would have a cusp point which is not practical for the purpose of a smooth curve design.

Quintic Hermite curves can be used to keep curvature continuous over the length of the curve. In the cubic case the curvature is usually discontinuous at the joins of consecutive curve sections.

### 5.3.4 Bezier Curve Interpolation

There are two candidates for Bezier interpolation of the points \( p_i \). The first is to define a Bezier curve with the \( p_i \) as control points. However, from the properties of Bezier curves in section 5.2.4 and the deductions of section 5.3.2, this curve
will not interpolate except at the end points. It may be possible to use an iterative process to refine the control points to ensure the curve passes through the points $p_i$, but the solution is not immediate.

The second candidate is similar in concept to Hermite interpolation. A cubic Bezier curve is defined between each consecutive pair of the $p_i$. Suppose $b_j, \quad 0 \leq j \leq 3n$ are the control points of the curve consisting of many cubic Bezier sections. Then set $b_{3i} = p_i, 0 \leq i \leq n$, leaving $b_{3i-1}$ and $b_{3i+1}$ as the intermediate control points. Then, from the result (5.12), for continuity in tangent, we require the following conditions for some $\alpha > 0$,

$$b_{3i} - b_{3i-1} = \alpha(b_{3i+1} - b_{3i})$$

Compare this with (5.37). Figure 5.18 shows an example in the cubic case. The last condition here is, in words, that the points $b_{3i-1}, b_{3i}, b_{3i+1}$ are collinear. The Catmull-Rom spline uses the tangent direction definition of (5.30), and places the control points at

Figure 5.18 For two Bezier curves to join smoothly requires a common end point and three collinear Bezier points.
If the Bezier segments were of higher order, intermediate control points would be estimated in a suitable manner, for example by placing them at the positions required to make the curve segment a degree-raised cubic Bezier curve. Curvature can be kept continuous, usually by using higher order Bezier curves and further conditions on the control points.

### 5.3.5 B-Spline Curve Interpolation

A first guess for interpolation with B-spline curves may be to try a B-spline curve with control points at the \( p_i \) and some knot sequence, perhaps uniform, perhaps generated by chord length, as suggested in section 5.3.7. However, as with the Bezier interpolation noted in 5.3.4, the B-spline curve will not interpolate the \( P_i \). It is possible to alter the control points so that the curve does pass through the \( P_i \). Lord [LORD 87] has suggested a method of iteratively refining the control points one by one until the curve passes through the \( P_i \). Barsky and Greenberg demonstrated that the positions for the control points could be solved algebraically to give an interpolatory curve [BARSKY & GREENBERG 80], and a concise and complete proof of this is given by Farin [FARIN 90]. A diagram of a resulting B-spline curve interpolation is given in Figure 5.19. B-spline curve interpolation has the significant advantages of B-spline curves once the control points of the B-spline curve are known.
5.3.6 Continuity

A note on the definition of continuity is required since many of the above curve representations and subsequent surface representations are piecewise polynomial curves and surfaces. The fact that a curve is piecewise polynomial means that the curve is comprised of many sub-sections of simply defined curves, as discussed in section 5.2. $C^n$ continuity of a curve over a parameter $t$ implies that the curve is $n$ times continuously differentiable over its range with respect to that parameter. This will require establishment of a parameter over the length of the whole curve. However, this project is not interested in the underlying parametrisation of the curve, but rather the shape of the curve. The term Geometric Continuity will be used, and $GC^n$ will denote Geometric Continuity of order $n$. A history of the term ‘Geometric Continuity’ is given by Goldman [GOLDMAN 90]. The lowest level of continuity is continuity of position, or $GC^0$. The next level is $GC^1$ denoting continuity of tangent in addition to $GC^0$, and the third level, $GC^2$ denotes continuity of curvature in addition to $GC^1$. In this project, the principle considerations are $GC^0$ and $GC^1$. 

Figure 5.19 B-spline curves can interpolate to points $p$, after control points $d_i$ have been determined.
These definitions of geometric continuity correspond to widely accepted definitions. For example, Farin [FARIN 90] uses the notation $G^n$ to denote \('n\)-times differentiable with respect to arc length parametrisation'. $G^0$ and $G^1$ agree with $GC^0$ and $GC^1$, although $G^2$ is stronger than $GC^2$ - it also requires continuity of the osculating plane. An alternative definition of a $G^2$ curve is that it is $C^2$ with respect to some parametrisation. Böhm uses the term Visual Continuity, which agrees with $GC^n$ for $n=0,1,2$ [BOHM 88], and Gregory uses the term Geometric Continuity of order $n$ to mean Frenet Frame differentiability of order $n$ [GREGORY 89]. Gregory’s definition agrees with Farin’s for $n=0,1,2$.

### 5.4 Surfaces in Three Dimensions

As discussed in the introduction to the current chapter, in three-dimensional computer aided design, curved surfaces are often generated from primitives, such as spheres and rectangular blocks, and extrusions and rotations of two-dimensional profiles. The class of primitives can include many shapes which vary in mathematical and physical complexity, and objects are formed by piecing together these primitives, extrusions and rotations. Sharp edges and joins are rounded or filletted to give overall smooth shapes. However, these methods are insufficient for modelling the surface shapes which are common in prosthetics and orthotics. This is because the shapes encountered are ‘doubly curved’, by which it is meant that the shape of one cross-section of the surface is not related to the shape of another cross-section of the surface except, possibly, in a superficial manner, where for example cross-sections in various directions may all be closed curves. The shapes available using the techniques are not flexible enough even to give good approximations to the shapes encountered unless an unacceptably high number of sections of simple primitives are used. Therefore, other techniques are required.

First attempts to develop new techniques were sponsored by the automotive and aircraft industries in order to design shapes which were aesthetically pleasing, which satisfied certain fluid dynamics properties and yet were not available from the common techniques already discussed. The earliest development of surfaces for use in design was by Coons, and the Coons patch
is named after him [COONS 67]. The Coons patch requires four boundary curves, and forms a smooth surface to interpolate these curves, and is described first in this section. The limitations of the Coons patch are first, when a collection of curves can be defined as the boundaries to a number of Coons patches, the joins between adjacent Coons patches will not, in general, be smooth, and second, Coons patches require boundary curves to be known. The latter limitation would require an extra stage in the application of a surface model since the general situation in prosthetics and orthotics is that a number of points on the surface is the only information that is known; here, the boundary curves would have to be determined before Coons patches could be developed. To overcome the first limitation, several parametric methods have been developed for representing surfaces. Some methods, such as tensor product B-spline surfaces, describe the entire surface with one definition while others, such as Hermite patches follow an alternative philosophy of dividing the surface up into a number of smaller portions or patches, each of which can be modelled and then imposing sufficient continuity conditions where the adjacent patches join. Many of the parametric methods are extensions of the curve methods presented in section 5.2, often to tensor product surfaces, and after discussion about the definition of a surface and the presentation of Coons patches, the definition of a tensor product surface and the common parametric methods are presented.

5.4.1 Definition of a Surface

Recall from section 5.2.1 that there were three possible forms for the definition for a curve. Analogously there are three possible forms for the definition of a surface within a Cartesian coordinate system:- Dependent Variable, Implicit and Parametric. Again, the form on which this project will concentrate is the Parametric definition, but all three forms are explained.

Dependent Variable
The dependent variable form, also known as Monge’s equation, is where one variable is dependent upon the other two, for example \( z \) is dependent on \( x, y \), or, mathematically expressed,
This form suffers from deficiencies similar to those for the dependent variable curve, namely that it is not suitable for surfaces which loop back on themselves or are closed surfaces, and the form is heavily dependent upon the orientation of the $x, y, z$ axes.

**Implicit Form**

The second form is the implicit definition, also known as the 'classical form', which is defined as being those points in space where some function is exactly zero, that is,

$$F(x, y, z) = 0$$

All surfaces can be represented in this manner, although they may be difficult to calculate and draw. An example of this form is a sphere, centre $(a, b, c)$, radius $r$, defined by

$$F(x, y, z) = (x - a)^2 + (y - b)^2 + (z - c)^2 - r^2 = 0$$

**Parametric Form**

The third form is the parametric definition form, also known as 'Gauss's form' and the 'explicit form', and is so called because the surface is defined in terms of two independent parameters. As the values of the parameters are varied over their allowed range, the defining equations yield every point on the surface. That is, the $x, y, z$ coordinates are functions of two independent parameters, $u, v$, or,

$$x = x(u, v)$$
$$y = y(u, v)$$
$$z = z(u, v)$$

The parameters $u, v$ are real variables which are often restricted in their allowed range, frequently $0 \leq u \leq 1$, $0 \leq v \leq 1$. For this project, a surface is defined as a
function determined by three such relationships in terms of two independent parameters, \(u, v\). A surface in this form is easily calculated at any point by substitution of the appropriate parameter values. As with curves, a reasonable sketch is often obtained by calculating the curve at ten equally spaced intervals in each of the parameters, and much better accuracy is achieved by use of a hundred equally spaced intervals. A simple impression of the surface can often be gained from the parametric form by substitution of a few values of each parameter, and this is a very useful feature for drawing a quick sketch by hand. One important point to note is that parameterisation of a surface is not unique. As an example, here are two different definitions of the sphere given above in implicit definition,

\[
x(u, v) = a + r \cos(2\pi u) \cos(2\pi v)
\]
\[
y(u, v) = b + r \sin(2\pi u) \cos(2\pi v)
\]
\[
z(u, v) = c + r \sin(2\pi v)
\]

and

\[
x(u, v) = a + r \cos(2\pi u^2) \cos(2\pi v)
\]
\[
y(u, v) = b + r \sin(2\pi u^2) \cos(2\pi v)
\]
\[
z(u, v) = c + r \sin(2\pi v)
\]

The surfaces presented in this section will all be in Parametric form.
5.4.2 Coons Patches

The bilinearly blended Coons patch, originally developed by Coons [COONS 67], interpolates to four given boundary curves giving a smooth patch between the curves. The four curves are treated in pairs. First a ruled surface is established between each pair by linear interpolation between the curves, and then a doubly ruled surface is established between the four corner points where the curves meet. A weighted average of these three surfaces gives the Bilinearly Blended Coons patch. However, the joining of two adjacent bilinearly blended Coons patches would not be smooth. This can be overcome by using more complicated blending techniques between the pairs of curves, but this requires the estimation of tangents across each boundary curve, and still there are problems.

Let the four boundary curves for a Coons patch be \( b_0(u), b_1(u), c_0(v), c_1(v) \), as shown in Figure 5.20, where \( 0 \leq u \leq 1, 0 \leq v \leq 1 \). The ends of adjacent curves are
assumed to be the same, otherwise there is a compatibility problem, and no smooth surface can be fitted. If the surface \( s(u,v) \) interpolates to the four corner points, then it will satisfy the conditions

\[
\begin{align*}
    s(0,0) &= b_0(0) = c_0(0) \\
    s(1,0) &= b_0(1) = c_1(0) \\
    s(0,1) &= b_1(0) = c_0(1) \\
    s(1,1) &= b_1(1) = c_1(1)
\end{align*}
\]

Similarly, if the surface \( s(u,v) \) interpolates the four boundary curves, then it will satisfy the conditions

\[
\begin{align*}
    s(u,0) &= b_0(u) \\
    s(u,1) &= b_1(u) \\
    s(0,v) &= c_0(v) \\
    s(1,v) &= c_1(v)
\end{align*}
\]

By linear interpolation of the opposite boundaries in the \( u,v \) directions respectively, the following two 'ruled' surfaces will be established (a 'ruled' surface is one formed by linear interpolation of two curves),

\[
\begin{align*}
    s_c(u,v) &= (1-u)c_0(v) + uc_1(v) \\
    s_h(u,v) &= (1-v)b_0(u) + vb_1(u)
\end{align*}
\]

Then the following bilinear surface (a 'bilinear surface' is a surface formed by two processes of linear interpolation) interpolates to the four corner points,

\[
s_{bc}(u,v) = (1-v)\{(1-u)p_{0,0} + up_{1,0}\} + v\{(1-u)p_{0,1} + up_{1,1}\}
\]

Then the surface \( s(u,v) \) defined by the following equation, and which is a weighted average of these three surfaces interpolates the four curves, and is known as a bilinearly blended Coons patch:
As stated above, two adjacent bilinearly blended Coons patches would not join smoothly. This is because the tangent across each boundary curve is dependent upon information not related to that boundary, and this renders the bilinearly blended Coons patch very limited in its applications. The general form of a Coons patch is obtained by replacing the blending functions for linear interpolation in the above derivation, \( u, (1-u), v, (1-v) \), by more general functions. The commonest method is to estimate the cross-boundary tangent at each point of the boundary curves, and to generate the 'bicubically blended' Coons patch. In the current text, the philosophy only is described and for further analysis the reader is referred to [FARIN 90]. In this case, the input data is not only the boundary curves, but also the derivatives

\[
\frac{\partial s}{\partial u}(0,v), \frac{\partial s}{\partial u}(1,v), \frac{\partial s}{\partial v}(u,0), \frac{\partial s}{\partial v}(u,1)
\]

The interpolation between each pair of opposite boundaries is formed by establishing a cubic Hermite curve as in (5.17) for each parameter value along the boundaries. The interpolation surface between the four corners involves the points and derivatives

\[
s(0,0), \frac{\partial s}{\partial u}(0,0), \frac{\partial s}{\partial v}(0,0), \frac{\partial^2 s}{\partial u \partial v}(0,0)
\]

with similar vectors for the other corners. For each value for both of the parameters a cubic Hermite interpolation is undertaken, yielding a surface. However, there is a subtle compatibility problem in this scheme. The value of the following twist vectors may not be unique since each may have been estimated twice, once from each of the boundary curves at the corner concerned,

\[
\frac{\partial^2 s}{\partial u \partial v}(0,0), \frac{\partial^2 s}{\partial u \partial v}(0,1), \frac{\partial^2 s}{\partial u \partial v}(1,0), \frac{\partial^2 s}{\partial u \partial v}(1,1)
\]

The resulting incompatibility was first noticed by Gregory, who proposed a
solution, known as Gregory's Square, which although often satisfactory, yields complications in calculation of the surface [GREGORY 74].

Despite these limitations, Coons patches can be used successfully in many circumstances, but when each of the boundary curves is a cubic curve and the twist vectors are unique, bicubic patches are the result, and the method yields the same surface as the subsequent tensor product Hermite patches of section 5.4.4, which are easier to understand.

5.4.3 Tensor Product Surfaces

In general, any curve method can be extended to a surface method by using a tensor product technique. Essentially, a curve can be swept through space, with each point on the curve sweeping through a curve in space. When the curves for each point are considered together, a surface is defined. The surface is known as a tensor product surface because of the nature of the defining equation. The most common form of tensor product surface is when the original curve and the curves along which each point is swept are of the same form. In this section, the derivation of the mathematical formula for a tensor product surface formed from a curve method defined over a number of control points is presented.

Consider \( c(u) \), a curve defined by basis functions \( F^*_i(u) \) over the control points \( c_i \) such that

\[
c(u) = \sum_{i=0}^{n} F^*_i(u)c_i
\]

If this curve is swept through three-dimensional space by changing the \( c_i \) in a continuous manner as it is swept, the curve can intuitively be seen to describe a surface. More precisely, if each \( c_i \) passes along a curve defined by basis functions \( G^*_i(v) \) over points \( p_{ij} \), ie

\[
c_i(v) = \sum_{j=0}^{n} G^*_j(v)p_{ij}
\]
then the resulting surface is a tensor product surface, so called because of the 
nature of the defining formula which appears as the product of two independent 
curve schemes. The property that the surface can be obtained from using two 
curve schemes is known as 'separability'. Mathematically the formula is

\[ c(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} F_i^*(u) G_j^*(v) p_{ij} \]  

(5.40)

Usually the basis functions are similar, and often of the same order. In this case, 
the equation of definition would be

\[ c(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} F_i^*(u) F_j^*(v) p_{ij} \]  

(5.41)

Note that the points \( p_{ij} \) are required for \( i = 0, \ldots, n; j = 0, \ldots, m \), and thus a 
complete rectangular structure of \((n + 1)(m + 1)\) points is required. Because the 
structure is rectangular, tensor product surfaces tend to be used to define 
approximately rectangular regions.

5.4.4 Tensor Product Bezier Patches

The Tensor Product approach to forming surfaces from curves is easily 
demonstrated with Bezier patches. Recall the definition of a Bezier curve (5.6),

\[ b(u) = \sum_{i=0}^{n} B_i^*(u) b_i \]

where the \( B_i^*(t) \) are the Bernstein polynomials of order \( n \). Using the derivation 
of tensor products in the previous section, let

\[ b_i(v) = \sum_{j=0}^{m} B_j^*(v) b_{ij} \]

then

\[ b(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_i^*(u) B_j^*(v) b_{ij} \]  

(5.42)
The $b_{ij}$ are the 'Bezier points' or 'control points' of the surface, Figure 5.21. At this stage, (5.8) is invoked to justify the formation of the tensor product Bezier patch: note that a point on the surface is obtained by a barycentric combination of points on Bezier curves, and each of these points is a barycentric combination of the Bezier points. As with Bezier curves, there are several notable properties of tensor product Bezier patches:

**Affine Invariance** Affine invariance is important because it implies that if all the Bezier points of a tensor product Bezier patch are subject to a rotation, translation or scaling, for example, then the surface will be subject to that same rotation, translation or scaling. The property follows from the preceding comment that a point on the surface is obtained by barycentric combinations of the control points.

**Convex Hull** The surface lies within the convex hull of all the control points.
This follows since

$$
\sum_{i=0}^{n} \sum_{j=0}^{m} B_i^m(u)B_j^n(v) = 1 \quad \text{for } 0 \leq u, v \leq 1
$$

(4.38)

**Corner Point Interpolation** When both of the parameters take extreme values 0 and 1, that is at the corners of the patch, the patch interpolates the corner control points

$$
\begin{align*}
\mathbf{b}(0,0) &= \mathbf{b}_{00} \\
\mathbf{b}(1,0) &= \mathbf{b}_{n0} \\
\mathbf{b}(0,1) &= \mathbf{b}_{0m} \\
\mathbf{b}(1,1) &= \mathbf{b}_{nm}
\end{align*}
$$

(5.43)

**End Curve Interpolation** The restriction to the values $u = 0, u = 1, v = 0, v = 1$ are Bezier curves which form the boundaries to the Bezier patch, defined respectively on $\mathbf{b}_{00}, \ldots, \mathbf{b}_{0m}, \mathbf{b}_{n0}, \ldots, \mathbf{b}_{nm}, \mathbf{b}_{00}, \ldots, \mathbf{b}_{n0}$ and $\mathbf{b}_{0m}, \ldots, \mathbf{b}_{nm}$.

**Symmetry** The surfaces defined on the Bezier points $\mathbf{b}_i$ with the following four sets of correspondancies are identical. In words, it does not matter from which corner the numbering of the control points starts - the surface will be the same for all corners.

\[
\begin{align*}
\begin{cases} 
  u = 0 \\
  u = 1 \\
  v = 0 \\
  v = 1
\end{cases}
\end{align*}
\begin{align*}
\begin{cases} 
  i = 0 \\
  i = n \\
  j = 0 \\
  j = m
\end{cases}
\end{align*}
\]

\[
\begin{align*}
\begin{cases} 
  u = 1 \\
  u = 0 \\
  v = 0 \\
  v = 1
\end{cases}
\end{align*}
\begin{align*}
\begin{cases} 
  i = 0 \\
  i = n \\
  j = 0 \\
  j = m
\end{cases}
\end{align*}
\]

This follows since

\[
\sum_{i=0}^{n} \sum_{j=0}^{m} B_i^u(u)B_j^v(v)b_{ij} = \sum_{i=0}^{n} \sum_{j=0}^{m} B_i^u(1-u)B_j^v(v)b_{n-i,j}
\]

\[
= \sum_{i=0}^{n} \sum_{j=0}^{m} B_i^u(u)B_j^v(1-v)b_{i,m-j}
\]

\[
= \sum_{i=0}^{n} \sum_{j=0}^{m} B_i^u(1-u)B_j^v(1-v)b_{n-i,m-j}
\]

**Intuitive Shape** The shape of the Bezier surface follows the shape of the polygon defined by the Bezier points in exactly the same way that the shape of a Bezier curve followed the shape of the polygon defined by its Bezier points in section 5.2.3. This is particularly useful for designing aesthetic shapes of surface.

**The de Casteljau Algorithm** The de Casteljau Algorithm for Bezier curves can be used for Bezier surfaces from the separability of the Bezier surface. The algorithm is fairly involved: if

\[
b_{ij}^{00} = b_{ij}
\]

\[
b_{ij}^{rs} = (1-u)b_{i-1,j}^{rs-1} + ub_{ij}^{rs-1} \quad 1 \leq r \leq i \leq n \quad (5.44)
\]

\[
b_{ij}^{sr} = (1-v)b_{i,j-1}^{sr-1} + vb_{ij}^{sr-1} \quad 1 \leq s \leq j \leq m
\]

then

\[
b(u, v) = b_{nm}^{nm} \quad (5.45)
\]
Note that the algorithm is driven by the superscripts, and the possible subscripts are limited by the superscripts. The algorithm can be followed in any order from a superscript $00$ to a superscript $nm$.

**Derivatives of a Tensor Product Bezier Surface** The derivatives of a tensor product Bezier surface can be written in terms of the Bernstein polynomials and the control points as

$$
\frac{\partial}{\partial u} b(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} \frac{\partial}{\partial u} B_i^n(u) B_j^m(v) b_{ij}
$$

$$
= \sum_{j=0}^{m} \left[ B_j^m(v) \frac{\partial}{\partial u} \left( \sum_{i=0}^{n} B_i^n(u) b_{ij} \right) \right]
$$

The term inside the square brackets can be simplified using the results of derivatives for Bezier curves in section 5.2.4 to give

$$
\frac{\partial}{\partial u} b(u, v) = \sum_{j=0}^{m} \left\{ B_j^m(v) n \sum_{i=0}^{n-1} B_i^n(u) \left[ b_{i+1,j} - b_{ij} \right] \right\}
$$

Indeed, the $r + s$-th derivative is given by

$$
\frac{\partial^{r+s}}{\partial u^r \partial v^s} b(u, v) = \frac{n!}{(n-r)!} \frac{m!}{(m-s)!} \sum_{i=0}^{n-r} \sum_{j=0}^{m-s} B_i^{n-r}(u) B_j^{m-s}(v) \Delta^r b_{ij}
$$

where

$$
\Delta^{00} b_{ij} = b_{ij}
$$

$$
\Delta^{rs} b_{ij} = \Delta^{r-1,s} b_{i+1,j} - \Delta^{r-1,s} b_{ij}
$$

$$
\Delta^{rr} b_{ij} = \Delta^{r-1,r} b_{i+1,j+1} - \Delta^{r-1,r} b_{ij}
$$

Note in particular that the $r$-th derivative across a patch boundary depends only on the first $r + 1$ rows of control points.
Figure 5.22 The deviation of the quadrilateral of control points near a corner of a Bezier patch is proportional to the twist at that corner.

The Twist of a Tensor Product Bezier Surface The twist of a surface $b(u,v)$ is defined as $\frac{\partial^2}{\partial u \partial v} b(u,v)$. The twist of the tensor product Bezier surface at its corner is

$$\frac{\partial^2}{\partial u \partial v} b(0,0) = mn \Delta^{11} b_{00}$$

$$= mn (b_{11} - b_{10} - b_{01} + b_{00})$$

The section of this last term within the brackets is the deviation of the quadrilateral $b_{00}, b_{01}, b_{11}, b_{10}$ from a parallelogram, Figure 5.22.

Tangent Plane of a Tensor Product Bezier Surface The tangent plane at a point on a surface $b(u,v)$ is specified by the vector which is normal to the plane. This vector is perpendicular to any vector in the tangent plane, and two vectors in
the tangent plane are \( \frac{\partial}{\partial u} b(u, v), \frac{\partial}{\partial v} b(u, v) \). Therefore, if these two vectors are linearly independent, which is true for most Bezier surfaces and all those encountered in this project, then the normal to the tangent plane at \( b(u, v) \) is given by

\[
\mathbf{n} = \text{unit} \left( \frac{\partial}{\partial u} b(u, v) \times \frac{\partial}{\partial v} b(u, v) \right) \tag{5.49}
\]

**Degree Elevation for a Tensor Product Bezier Surface** A Bezier patch of order \( m \) by \( n \) can be raised to a Bezier patch of order \( m + 1 \) by \( n \) by degree elevation of each of the rows of Bezier points using the same procedure as for elevation of a Bezier curve (5.15). Similarly the patch could be raised to a Bezier patch of \( m \) by \( n + 1 \) by degree elevation of each of the columns of Bezier points. Using a combination of these two elevations, the patch can be raised to any order \( m + r \) by \( n + s \) with \( r \geq 0, s \geq 0 \).

**Variation Diminishing** There is no extension of the term 'Variation Diminishing' to surfaces. It is not clear what the term would mean.

**Avoidance of Looped Surfaces** A looped surface is a surface which passes through some point twice. It might be hoped to be able to establish a condition in terms of the Bezier points for the avoidance of loops in tensor product Bezier surfaces. However, as with variation diminishing, this property cannot be generalised from curves to surfaces. It would seem wise to ensure that each row and each column of Bezier points should be monotonic in a direction which will be different for rows and for columns, but this does not turn out to be a guarantee that there are no loops.
Forward Difference Calculation Forward Difference calculation techniques carry over directly from the curve case to the tensor product surface case. The positions of points on the surface would be calculated by rows. The order of the points calculated would read across the following table,

\[
\begin{array}{cccccc}
\mathbf{b}(0,0) & \mathbf{b}(0,\mathbf{h}) & \mathbf{b}(0,2\mathbf{h}) & \ldots & \mathbf{b}(0,1) \\
\mathbf{b}(d,0) & \mathbf{b}(d,\mathbf{h}) & \mathbf{b}(d,2\mathbf{h}) & \ldots & \mathbf{b}(d,1) \\
\mathbf{b}(2d,0) & \mathbf{b}(2d,\mathbf{h}) & \mathbf{b}(2d,2\mathbf{h}) & \ldots & \mathbf{b}(2d,1) \\
\ldots & \ldots & \ldots & \ldots & \ldots \\
\mathbf{b}(1,0) & \mathbf{b}(1,\mathbf{h}) & \mathbf{b}(1,2\mathbf{h}) & \ldots & \mathbf{b}(1,1) \\
\end{array}
\]

5.4.5 Tensor Product Hermite Patches

A tensor product Hermite patch is created by passing each point of a Hermite curve along another Hermite curve (see section 5.2.4). The total amount of information for a cubic tensor product Hermite patch is the position, derivative with respect to each of the parameters and the twist at each corner. The vectors \( \mathbf{p}_{00}, \mathbf{p}_{u0}, \mathbf{p}_{v0}, \mathbf{p}_{uv00} \) are, respectively, the position, the derivative with respect to \( u \), the
derivative with respect to \( v \) and the twist vectors at the corner \( u = v = 0 \). Similarly for the other corners, see also Figure 5.23. Recall the definition of a Hermite cubic curve (5.17)

\[
h(u) = (1 - u)^2 (1 + 2u)a + (3 - 2u)u^2 b + (1 - u)^2 us - u^2(1 - u)t
\]

In matrix terms, from (5.28), this is

\[
h(u) = (a b s t) M [u]
\]

where \([u] = (1 \ u \ u^2 \ u^3)^T\), and

\[
M = \begin{pmatrix}
  1 & 0 & -3 & 2 \\
  0 & 0 & 3 & 2 \\
  0 & 1 & -2 & 1 \\
  0 & 0 & -1 & 1 \\
\end{pmatrix}
\]

Now let

\[
a = a(v) = (P_{00} \ P_{01} \ P_{v0} \ P_{v1}) M [v]
\]

\[
b = b(v) = (P_{10} \ P_{11} \ P_{v10} \ P_{v11}) M [v]
\]

\[
s = s(v) = (P_{000} \ P_{0v0} \ P_{vv0} \ P_{vv1}) M [v]
\]

\[
t = t(v) = (P_{u0} \ P_{u1} \ P_{uv0} \ P_{uv1}) M [v]
\]

where \([v] = (1 \ v \ v^2 \ v^3)^T\). Then the tensor product Hermite patch is given by

\[
h(u, v) = [u]^T M^T \begin{pmatrix}
  P_{00} & P_{01} & P_{v0} & P_{v1} \\
  P_{10} & P_{11} & P_{v10} & P_{v11} \\
  P_{u0} & P_{u1} & P_{uv0} & P_{uv1} \\
  P_{u10} & P_{u11} & P_{uv10} & P_{uv11}
\end{pmatrix} M [v] \tag{5.50}
\]

It was noted in section 5.2.4 that the Hermite cubic curve was a cubic Bezier curve, and the defining characteristics were given in terms of the Bezier points.
Using the previous results of section 5.2.4, noting that \( h(u,0), h(0,v) \) are cubic Bezier curves and using the derivative results of section 5.4.4 for Bezier surfaces, \( h(u,v) \) is a bicubic tensor product Bezier surface with

\[
\begin{align*}
  p_{00} &= b_{00} \\
  p_{u0} &= 3(b_{10} - b_{00}) \\
  p_{v0} &= 3(b_{01} - b_{00}) \\
  p_{uv0} &= 9(b_{11} - b_{01} - b_{10} + b_{00})
\end{align*}
\]

with similar conditions at the other corners.

### 5.4.6 Tensor Product B-Spline Surfaces

As with Bezier surfaces, B-Spline curves can be used to generate tensor product surfaces. The definition is

\[
p(u,v) = \sum_{i=0}^{i+k-1} \sum_{j=0}^{j+k-1} N_i^k(u)N_j^k(v)b_{ij}
\]

The orders of the two sets of basis functions do not have to be the same, \( k \), but in most practical cases they are. The tensor product B-spline surface requires a set of control points \( b_{ij} \) and two knot sequences \( ... \leq u_{i-1} \leq u_i \leq u_{i+1} \leq ... \) and \( ... \leq v_{j-1} \leq v_j \leq v_{j+1} \leq ... \).

Tensor product B-spline surfaces inherit many properties from B-spline curves, namely the convex hull property, differentiability, intuitive shape and local control. The De Boor algorithm can also be established. Note that with knot insertion a whole line of knots is introduced, occurring in the knot sequence for each row of control points. One other important property is that B-Spline surfaces are composite tensor product Bezier surfaces, and the derivation of this fact follows directly from the analogous curve result.
5.4.7 Matrix Representation of a Tensor Product Surface

As with curves, any parametric polynomial tensor product surface, of which tensor product Bezier, Hermite and B-spline surfaces are examples, can also be expressed in matrix form in terms of the bases \( u^0, u^1, u^2, u^3, ..., u^n \) and \( v^0, v^1, v^2, v^3, ..., v^n \). The matrix expression takes the form,

\[
p(u,v) = (u^0 u^1 ... u^n) M (p_00 ... p_{mn})^T \begin{pmatrix} t^0 \\ \vdots \\ t^n \end{pmatrix}
\]

For a tensor product Bezier surface, the matrix \( M \) will depend only upon the order of the curve. For a bicubic Bezier surface, the equation is

\[
b(u,v) = (1 u u^2 u^3) \begin{pmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{pmatrix} \begin{pmatrix} b_{00} & b_{10} & b_{20} & b_{30} \\ b_{01} & b_{11} & b_{21} & b_{31} \\ b_{02} & b_{12} & b_{22} & b_{32} \\ b_{03} & b_{13} & b_{23} & b_{33} \end{pmatrix} \begin{pmatrix} 1 \\ -3 \\ 3 \\ -1 \end{pmatrix}
\]

\[
\begin{pmatrix} \phi^0 \\ \phi^1 \\ \phi^2 \\ \phi^3 \end{pmatrix}
\]

(5.52)

Similarly, for the uniform cubic B-spline surface, with \( \theta = u - i - 3 \) and \( \phi = v - j - 3 \),

\[
b(u,v) = (1 \theta^1 \theta^2) \frac{1}{6} \begin{pmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{pmatrix} \begin{pmatrix} b_{,j} & b_{,j+1} & b_{,j+2} & b_{,j+3} \\ b_{,j+1,j} & b_{,j+1,j+1} & b_{,j+1,j+2} & b_{,j+1,j+3} \\ b_{,j+2,j} & b_{,j+2,j+1} & b_{,j+2,j+2} & b_{,j+2,j+3} \\ b_{,j+3,j} & b_{,j+3,j+1} & b_{,j+3,j+2} & b_{,j+3,j+3} \end{pmatrix} \begin{pmatrix} 1 \\ -3 \\ 3 \\ -1 \end{pmatrix}
\]

\[
\begin{pmatrix} \phi^0 \\ \phi^1 \\ \phi^2 \\ \phi^3 \end{pmatrix}
\]

(5.53)

The Hermite cubic patch was presented in its matrix form. Matrix form can sometimes be used to give efficient calculation, particularly if matrix multiplication is hard-wired. However, matrix manipulation with curves in this manner can be numerically unstable because of the large number of multiplications required.
5.4.8 Bezier Triangles

Although tensor product surfaces are the most common, there are some other possibilities for surfaces, particularly Bezier Triangles, and although they will not be used subsequently in this project, a discussion of parametric surface methods is not complete without their mention. Bezier triangles are an alternative extension of Bezier curves to surfaces, and were the first case considered by De Casteljau [DECASTELJAU 63]. For a complete description and discussion the reader is referred to an alternative review, for example [FARIN 90, BÖHM et al 84].

Consider a plane defined by the points \(a, b, c\). Then any point \(p\) in the plane can be written as a barycentric combination of these points, where a barycentric combination is a weighted average, such that the sum of the weights is 1, ie

\[
p = ua + vb + wc
\]

where \(u + v + w = 1\). The Bezier polynomials over values of \(u, v, w\) which sum to
1 are defined by

\[ B_{ijk}^n(u, v, w) = \frac{n!}{i!j!k!} u^i v^j w^k \]

where \( i, j, k \geq 0, i + j + k = n \). The Bernstein polynomials have many properties similar to the properties of the previously discussed polynomials in section 5.2.3, such as partition of unity, positivity, recursion.

A Bezier Triangle of order \( n \) is defined over Bezier points \( b_{ijk} \) where \( i + j + k = n, i, j, k \geq 0 \), with the parameter values restricted such that \( u, v, w \geq 0, u + v + w = 1 \) as

\[ b(u, v, w) = \sum S b_{ijk} B_{ijk}^n(u, v, w) \] (5.54)

where \( S \) denotes the sum over all terms where \( i, j, k \geq 0, i + j + k = n \). An example is shown in Figure 5.24. A De Casteljau algorithm can be established for this triangle and degree elevation and subdivision properties can be established similar to those for Bezier curves.

### 5.5 Interpolation with Surfaces in a Rectangular Situation

In the previous section, various parametric surfaces have been defined. When the surfaces are used in interpolation situations, it becomes apparent that many of them are ideally suited to one particular situation - a 'rectangular' problem. A rectangular problem is a situation where the points to be interpolated lie in rows and columns, and each point can be uniquely specified by a row number and a column number. The points can be thought of as lying at the vertices of a rectangular grid. Because of their natural use in such situations, the rectangular problem is the commonest situation to have been applied, for example, in car, ship and yacht design [BARSKY & GREENBERG 80, CATLEY 90, REUDING 89], in the shoe industry [GUNDILL & HACKNEY 88], for hydrographic survey [MACCARTHY & HANDSCOMB 89], knee-replacement
design [WALKER 88], and surface replication [VERGEEST et al 87]. The methods of section 5.4 are applied to a rectangular problem in this section and the more general case is discussed in section 5.6.

5.5.1 The Interpolation Requirements

A typical challenge for a rectangular situation requiring surface interpolation is to find a curve which passes through points \( p_{i,j} \), \( 0 \leq i \leq m; 0 \leq j \leq n \) in the order signified by their numbering. At this stage, no restrictions are imposed upon the relative distances between the rows of points, although all the methods will give more aesthetically pleasing shapes if the distances between the points in each pair of consecutive rows or columns are fairly similar. Many of the methods previously mentioned give only one patch between four adjacent points, and so many patches will have to be joined together to form a surface which interpolates all the points. Others will be able to form a curve through all the points with only one definition.

In section 5.4, the surfaces which were available for designing shape were in parametric form. For these definitions to be applicable to the current interpolation problem with several patches, the whole surface should be definable over two global parameters, and there will need to be a value \( u_i \) for each row of points and a value \( v_j \) for each column of points in the \( p_{ij} \) such that the surface passes through the point \( p_{ij} \) at the parameter value \((u_i, v_j)\). Often these parameter values are not known, and so methods are given for their estimation.

Some of the surfaces which have been suggested will require the tangent vectors corresponding to the \( u \) and \( v \) directions and also the twist vector at each of the points \( p_{ij} \). Methods for estimating these tangent and twist values are given for those cases in which such estimation is necessary. After the parameter, tangent and twist estimation methods, applications of the various methods of section 5.4 are given, followed by remarks upon the limitations of the rectangular problem.
Parameter Value Estimation

Parameter values \((u_i, v_j)\) for each of the points \(p_{ij}\) are often not known, but are required by the surface representation used. Estimation methods are therefore required. The simplest method is to set

\[
\begin{align*}
  u_i &= i \text{ for } 0 \leq i \leq m \\
  v_j &= j \text{ for } 0 \leq j \leq n
\end{align*}
\]

A second method is to estimate the difference in parameter value between successive rows as the average chord length between the points in the two rows, ie

\[
\begin{align*}
  u_0 &= 0 \\
  u_{i+1} &= u_i + \frac{1}{n} \sum_{j=0}^{n-1} \| p_{i+1,j} - p_{ij} \|
\end{align*}
\]

and similarly for the \(v_j\). However, if the points are not fairly evenly spaced, ie if for any \(i\)

\[
\max_j \{ \| p_{i+1,j} - p_{ij} \| \} \gg \min_j \{ \| p_{i+1,j} - p_{ij} \| \}
\]

then strange results, such as loops, can occur in the surface shape. Examples of this feature are given by Farin and Foley [FARIN 90, FOLEY 90]. In such a case the solution may be to use 'rational' surfaces, which are an extension of the simple curve schemes suggested in section 5.4. For a complete discussion of rational curves and surfaces, the reader is referred to a more general survey eg [FARIN 90, BOHM et al 84] and for rational B-splines to [PIEGL 89a, PIEGL 89b].
Tangent Estimation

It may be that tangent directions are specified at the points $p_i$. If not then the tangent directions often require estimation. The most common method is to estimate $\frac{\partial}{\partial u} S(u, v)$ to be the tangent direction generated by considering a curve through points $p_{i-1, j}, p_{i, j}, p_{i+1, j}$ at parameter values $u_{i-1}, u_i, u_{i+1}$ and to use one of the tangent estimation methods given in section 5.3.1. The derivative $\frac{\partial}{\partial v} S(u, v)$ is estimated in a similar manner.

Twist Estimation

As was most obviously demonstrated by the tensor product Hermite approach, position and tangent information is insufficient for the design of a surface. Twist information at the points, $\frac{\partial^2}{\partial u \partial v} S(u, v)$, is often required. There are various techniques for twist estimation, some of which are now given. More complete surveys are [BARNHILL et al 88, FARIN 90].

The first possibility is to set all twist values to be zero. Zero twists are mathematically acceptable and give rise to smooth surfaces. However, zero twists produce 'flat spots' at the points where zero twists are specified. This feature appears as unnaturally flat regions round each of the points $p_i$, and these surfaces are generally regarded as unacceptable [FARIN 90, BARNHILL et al 88].

A second scheme, known as Adini's twist, was first introduced in [BARNHILL et al 78] and is based on a finite element procedure known as 'Adini's Rectangle'. The scheme is to estimate the twist as
\[
\frac{\partial^2}{\partial u \partial v} S(u_i, v_j) = \frac{\partial}{\partial v} S(u_{i+1}, v_j) - \frac{\partial}{\partial v} S(u_{i-1}, v_j) + \frac{\partial}{\partial u} S(u_i, v_{j+1}) - \frac{\partial}{\partial u} S(u_i, v_{j-1}) + \frac{S(u_{i+1}, v_{j+1}) - S(u_{i-1}, v_{j+1}) - S(u_{i+1}, v_{j-1}) + S(u_{i-1}, v_{j-1})}{(u_{i+1} - u_{i-1})(v_{j+1} - v_{j-1})}
\]

\[S(u_{i+1}, v_{j+1}) - S(u_{i-1}, v_{j+1}) - S(u_{i+1}, v_{j-1}) + S(u_{i-1}, v_{j-1})
\]

A third possibility, Bessel's twist, is very similar in idea to Bessel's tangent discussed in section 5.3.1. The twist at \( p_{ij} \) is the twist to the biquadratic surface through the points \( p_{i+k,j+l} \) where \( k, l \in \{-1, 0, 1\} \). Let

\[
q_{ij} = \frac{1}{(u_{i+1} - u_i)(v_{j+1} - v_j)} \{ p_{i+1,j+1} - p_{i+1,j} - p_{i,j+1} + p_{ij} \}
\]

Then, with simplification after Farin [FARIN 90],

\[
\frac{\partial^2}{\partial u \partial v} s(u_i, v_j) = (1 - \alpha_i \alpha_j) \begin{pmatrix} q_{i-1,j-1} & q_{i-1,j} \\ q_{i,j-1} & q_{ij} \end{pmatrix} \begin{pmatrix} 1 - \beta_j \\ \beta_j \end{pmatrix}
\]

where \( \alpha_i = \frac{u_i - u_{i-1}}{u_{i+1} - u_{i-1}} \) and \( \beta_j = \frac{v_j - v_{j-1}}{v_{j+1} - v_{j-1}} \).

### 5.5.2 Interpolation with Coons Patches

Coons patches, as described in section 5.4.2 can be implemented to interpolate to the given \( p_{ij} \), with one patch between each set of points \( p_{ij}, p_{i+1,j}, p_{i+1,j+1}, p_{i,j+1} \) if curves have already been defined between the points. If not, they could be estimated by cubic Bezier curves, for example. The bilinearly blended Coons patch will then interpolate to the four boundary curves. However, there may not be tangent plane continuity across the boundaries between adjacent patches. This can be overcome if the cross-boundary tangent is specified along each boundary curve and a more complicated Coons patch, such as the
bicubically blended Coons patch, is used, but these specifications are time consuming, and in this situation where only the positions of the $p_y$ are known it is easier to use one of the subsequent methods.

### 5.5.3 Hermite Interpolation

Hermite bicubic patches can be used to interpolate to the given $p_y$ with one patch between each set of points $p_y, p_{i+1,j}, p_{i+1,j+1}, p_{i,j+1}$. Setting $u^* = \frac{u - u_i}{u_{i+1} - u_i}$, $v^* = \frac{v - v_j}{v_{j+1} - v_j}$, this gives

$$\frac{\partial}{\partial u} s(u,v) = \frac{du^*}{du} \frac{\partial}{\partial u} s(u,v)$$

$$= \frac{1}{(u_{i+1} - u_i) \frac{\partial}{\partial u} s(u,v)}$$

Thus for the Hermite patch with corner points $p_y, p_{i+1,j}, p_{i+1,j+1}, p_{i,j+1}$ and using the notation of (5.45),

$$p_{00} = p_y$$

$$p_{u0} = (u_{i+1} - u_i) \frac{\partial}{\partial u} S(u,v_j)$$

$$p_{v0} = (v_{j+1} - v_j) \frac{\partial}{\partial v} S(u,v_j)$$

$$p_{w0} = (u_{i+1} - u_i)(v_{j+1} - v_j) \frac{\partial^2}{\partial u \partial v} S(u,v_j)$$

Similar conditions are required at the other corners. Recall that in the curve case in section 5.2.4, biquintic Hermite curves could be used to interpolate additionally to curvature at its ends. Similarly, biquintic Hermite surface patches could be used to obtain continuity of curvature in addition to tangent.
(The order has gone up from bicubic to biquintic because there are two extra conditions in the direction of each parameter, namely the curvature at each edge of the patch.)

5.5.4  Tensor Product Bezier Interpolation

As with Bezier curve interpolation in section 5.3.4, there are two possibilities for interpolation with tensor product Bezier surfaces. The first possibility is a global solution with Bezier points given by \( b_{ij} = p_{ij} \). However, this does not interpolate to the Bezier points except at the corners, and moreover, the surface is everywhere affected by each Bezier point. For these reasons this possibility is rejected. Compare this with the first case of Bezier curve interpolation, section 5.3.4

The second possibility is in fact identical to the Hermite interpolation scheme in section 5.5.3. A tensor product Bezier patch is used to interpolate to each set of points \( p_{ij}, p_{i+1,j}, p_{i+1,j+1}, p_{i,j+1} \). Commonly the patch is bicubic, but more generally the patch may be of higher order than bicubic, say \( m \) by \( n \). From (5.43) the tangent and twist conditions require

\[
\begin{align*}
    b_{00} &= p_{ij} \\
    b_{01} &= b_{00} + \frac{1}{n} (u_{i+1} - u_i) \left( \frac{\partial}{\partial u} S(u_i, v_j) \right) \\
    b_{10} &= b_{00} + \frac{1}{m} (v_{j+1} - v_j) \left( \frac{\partial}{\partial v} S(u_i, v_j) \right) \\
    b_{11} &= b_{00} + (b_{10} - b_{00}) + (b_{01} - b_{00}) + \frac{1}{mn} (u_{i+1} - u_i) (v_{j+1} - v_j) \left( \frac{\partial^2}{\partial u \partial v} S(u_i, v_j) \right)
\end{align*}
\] (5.59)

with similar conditions at the other corners.
Consider two adjacent patches \( b^{(1)}(u, v), b^{(2)}(u, v) \), adjacent along the parameter \( u = u_i \). There are \( m \) Bezier points along the common boundary of the patches, and the above scheme can be seen to ensure that the triplets of points across the boundary are related, as in Figure 5.25, by

\[
(u_{i+1} - u_i) [b^{(2)}_{j,1} - b^{(2)}_{j,0}] = (u_i - u_{i-1}) [b^{(1)}_{j,m} - b^{(1)}_{j,m-1}] \quad \text{for} \quad j = 0, 1, m - 1, m \tag{5.60}
\]

If this relationship is ensured for all values \( 0 \leq i \leq m \) then the boundary between the two patches will be tangent continuous. For a cubic patch this implies no extra conditions. Around the vertices \( p_{ij} \) the tangent plane is well defined because each of the four patches at a vertex is normal to the plane \( \frac{\partial}{\partial u} S(u_i, v_j) \times \frac{\partial}{\partial v} S(u_i, v_j) \). Curvature continuity could be ensured by use of quintic or higher order tensor product Bezier patches.
Figure 5.26 Tensor product B-splines can interpolate points $p_{ij}$, but this requires determination of the control points $d_{ij}$.

Note that with cubic Hermite tensor product patches, specification of the position and tangents at the corners was insufficient to define the entire patch - twist information was also required. For tensor product cubic Bezier patches, the control points on the edge curves of each patch are insufficient to define the entire patch. The central points are also required. However, a suitable approach to constructing the surface can be to establish the Bezier points for the curves along the edges of the patches, and then to 'fill in' the points required in the interior of the patches.

5.5.5 Tensor Product B-Spline Interpolation

The method for tensor product B-spline interpolation is the most common technique found where the problem is of a rectangular nature, especially in industry. From the list given at the introduction to this section, the following use B-splines: [BARSKY and GREENBERG 80, CATLEY 90, REUDING 89, MACCARTHY & HANDSCOMB 89, WALKER 88, VERGEEST et al 87]. The
method is similar to curve B-spline interpolation as discussed in section 5.3.5. A first guess might be to set the control points at the $p_i$ with a suitable knot sequence, possibly uniform, possibly with a knot for each given $u_i, v_i$. Note that the same knot sequence in $u$ is required for each row of control points, and the same knot sequence in $v$ for each column of control points. However, this B-spline surface will not interpolate the $p_i$. It is possible to alter the control points so that the surface does pass through the $p_i$. Lord’s method [LORD 87] could be developed to iteratively refine each control point individually until the surface passes through the $p_i$. However, the positions of the control points can be algebraically determined so that the surface interpolates the $p_i$. Barsky has given a lengthy derivation of the positions for the control points [BARSKY & GREENBERG 80], and others have observed that there is a much shorter derivation for the positions of the control points, for example [FARIN 90]. A diagram of a B-spline interpolation is given in Figure 5.26. Note in particular that this method cannot be extended to non-rectangular problems.

An alternative method has been suggested for generation of surfaces by recursion. A control point mesh is used to produce a finer mesh, and this process is repeated. The final result is a B-spline surface, as shown by Catmul and Clark [CATMUL & CLARK 78]. Their paper claims not to be restricted to rectangular problems, but close examination reveals extraordinary points where the mesh is not rectangular. Similar schemes have been suggested by de Boor and by Gregory and Qu [DEBOOR 87, GREGORY & QU 88].

5.5.6 Continuity

The definitions of continuity in surface cases are generalisations of those given in section 5.3.6 for the curve case. Many surface schemes involve the piecing together of various patches of surface elements of types mentioned in section 5.4. Some terminology is required for specification of the continuity of the surface at the joins of the various patches. In this project, the notation $GC^n$ is used to mean ‘Geometric Continuity of Order $n$’. The lowest order of continuity is $GC^0$, denoting continuity in position. The next order, which will be the one concentrated upon for this project, is $GC^1$ denoting continuity in tangent plane.
Geometric Continuity of Order 2, $GC^2$ denotes continuity in principal curvatures. Other definitions of continuity are direct extensions of the curve continuity definitions in section 5.3.6; for example, $G^n$, signifies that the surface is locally $C^2$ with respect to some parametrisation [FARIN 90], $VC^n$, that the surface is visually continuous of order n [PIPER 87], and $GC^n$, that the surface is Frenet-Frame continuous of order n [GREGORY 89]. They all agree with the definition of Geometric Continuity for orders 0 and 1, and are slightly stronger for order 2.

5.5.7 Restrictions of the Rectangular Situation

To conclude this section we discuss in more detail the restrictions of the rectangular problem, and show that the problem is not applicable to a general closed object. In this context, a closed object is an object which surrounds and contains a volume of space and can be thought of as an object which could contain a liquid in three dimensional space, and whichever way the object was held the liquid would be unable to escape. The rectangular layout of the points $p_i$ can alternatively be viewed as the points lying on the vertices of a mesh with each vertex lying on four edges (ie connected to four other vertices), and each face of the mesh being enclosed by four edges. This also means that each edge separates two faces and each face touches four vertices. Therefore, if the number of vertices, edges and faces are $V, E, F$ respectively, then

$$4V = 2E$$
$$4F = 2E$$

Therefore,

$$V + F - E = 0$$

(5.61) is Euler’s equation. Any standard text on graph theory, eg [BOLOBAS 81], will show that for a closed body containing no 'holes' or 'handles' (that is a simple closed body like a sphere, a cube, as opposed to a torus or a cup),
Therefore, the rectangular problem cannot be applied to a closed body without holes or handles. (5.61) is in fact Euler's equation for a torus, and a little thought shows that the rectangular problem can be applied in two other cases, both restrictions of the torus, namely a cylinder and a sheet. These three possibilities are shown in Figure 5.27. Any of these shapes can, of course, be highly distorted. Hence the fairly wide range of possible applications.

5.6 Interpolation with Surfaces in a Non-Rectangular Situation

Several surface modelling techniques were developed in section 5.4, and in section 5.5 one particular interpolation situation was solved using these techniques. The problem was the simplest for solution, and the one most commonly found in the literature, but it was only suitable for interpolation of a set of points $p_{ij}$ which formed a rectangular array. Section 5.5.7 showed that the
only situations which can form a rectangular problem are distortions of a flat sheet, a cylinder and a torus. The solution is not, therefore, generally applicable, and other solutions are required for different situations. In this section, examples of other interpolation situations and solutions are given. None of the common methods will be found to be suitable for general use in prosthetics and orthotics, and further development will be required which will be presented in the next chapter.

5.6.1 Shephard’s Methods

Shephard’s methods are a dependent variable technique, and establish a function \( z = s(x, y) \) which interpolates to given functional values, that is it assumes given values at known points, or \( s(x_i, y_i) = z_i \) for \( 1 \leq i \leq n \). The class of methods is named after Shephard who first discussed a solution to this problem.
[SHEPHARD 65]. It is reasonable to demand that the effect of the value $z_i$ diminish as distance $d_i$ from $(x,y)$ to $(x_i,y_i)$ increases. Such a function, which ensures slope continuity everywhere is

$$s(x, y) = \frac{\left( \sum_i \frac{z}{d^2} \right)}{\left( \sum_i \frac{1}{d^2} \right)}$$

This function in fact has flat spots at each interpolation point. The function can be improved so that it also interpolates to an estimated tangent direction at each point [BARNHILL 77]. Similar techniques including ‘radial basis functions’, that is techniques which ensure that the effect of a point is equal in all directions from the point, are reviewed by Franke and Jackson [FRANKE 82, JACKSON 89]. The restriction of such methods is that the surface is in dependent variable form, and therefore, there must be a plane domain over which the function can be defined for all the data points. Shephard’s methods are most readily applicable in geography and geology, where the local portion of the earth's surface can be treated as a height function over a plane [MACCARTHY & HANDSCOMB 89].

Shephard’s methods are not applicable in most prosthetics and orthotics applications, because there is no plane over which the surface is a univalued function, and so are rejected for this project. It may be possible to use a function defined over some other domain, but this is a new research topic in mathematics and beyond the scope of the current project [FOLEY 90, BARNHILL 89]. Moreover, if the domain is non-convex then there are difficulties in determining which points in space are related to which points in the domain. A non-convex domain would be required for a complete foot model, for example, as shown in Figure 5.28.
Figure 5.29 A regular triangular mesh for Farin's method.

Figure 5.30 An arbitrary triangular mesh which requires Piper's method.
5.6.2 Triangular Patch Methods

The commonest alternative structure to the use of tensor product patches is the use of the triangular Bezier patches of section 5.4.7. The earliest such solution requires a regular hexagonal mesh of points to be interpolated [FARIN 82]. In effect, this requires the overall shape to be a (distorted) flat sheet, cylinder or torus, the possibilities outlined in section 5.5.8 for the rectangular problem, and an example mesh is given in Figure 5.29. A general solution to the problem of establishing a Bezier triangle surface through points \( p_i \) is given by Piper [PIPER 87], and an example is given in Figure 5.30. Each of the points \( p_i \) is associated with a vertex of a mesh, each face of which is a triangle - these points are referred to as having been 'triangulated'. Piper's solution requires only a previously established triangulation of the points and a tangent plane at each point. It is therefore generally applicable. Its major disadvantage over the methods of the following two sections is that it is more difficult to establish a triangulation than a largely rectangular mesh.
5.6.3 Largely Rectangular Situation Part 1 - Four Patch Vertices

In section 5.5.8 the rectangular situation was restated as points \( p_i \), associated with the vertices of a mesh such that each vertex is attached to four edges and each face is enclosed by four edges. The first possible relaxation of this situation is to require that most of the points satisfy these conditions, but some faces of the mesh are allowed to have other numbers of sides. However, each point must still be adjacent to four other points. Examples of the types of surfaces which can now be modelled are given in Figure 5.31. Suppose that there are \( r \) faces with three edges, \( s \) faces with five edges, and all other faces with four edges.

Then

\[
3r + 5s + 4(F - r - s) = 2E
\]

\[
4V = 2E
\]

To satisfy Euler’s equation for a closed object,

\[
V + F - E = 2
\]

and this requires

\[
r = s + 8
\] \hspace{1cm} (5.64)

The simplest solution to this is the Octahedron, with \( V = 6, F = 8, E = 12, r = 8, s = 0. \)

A significant advantage of the approach of this section is that tangent estimation methods discussed in section 5.5.1 are still applicable throughout the surface, and the twist estimation methods require only minor modifications near the non four-sided patches. A solution to the problem will typically use methods of section 5.5 to estimate all areas of the surface, except within the non four-sided patches. Gregory and Charrot [GREGORY & CHARROT 80, CHARROT & GREGORY 84] suggested a three-sided patch for solution to this type of
problem, and Gregory has also suggested a five-sided patch [GREGORY 86]. Triangular Bezier patches are an alternative possibility, and Box-Splines [DAEHLLEN & SKYTT 87] are a method for establishing a solution for three-, five- and six-sided patches. Further suggestions are made by Gregory and Hahn, and DeRose and Loop [GREGORY & HAHN 87, DEROSE & LOOP 88]. A disadvantage of all of these methods is that they require a different method for surface calculation in the non four-sided patches, and hence a different computer algorithm for their implementation.

5.6.4 Largely Rectangular Situation Part 2 - Four Sided Patches

An alternative relaxation to the rectangular problem to that given in section 5.6.3 is to let some vertices of the mesh be attached to a number other than four other edges. Examples of the meshes which can then be generated are given in Figure 5.32, and it is again easy to see that many shapes can be modelled. The advantage of this approach is that it is conceivable that all patches can be of the same type,
such as those suggested in section 5.4. There are, however, some other parts of the process for which alternative techniques are required. At the irregular points, the tangent directions cannot be estimated by the techniques in section 5.5.1. Curve techniques from section 5.3.1 can be used once the tangent plane at the irregular points has been established. A method of establishing the tangent plane at each irregular point will be required. Similarly, techniques for estimating the twist of the surface at the irregular points will be required.

Following a similar analysis to that given in the previous section, if there are $r$ vertices attached to three edges, $s$ attached to five edges, and all others attached to four, then to satisfy (5.62),

$$r = s + 8$$

(5.65)

It is no surprise to see that the cube is the simplest possible situation, with eight vertices, each attached to three edges.

The mathematical solution for the surface around the irregular vertices is not as simple as that at the regular vertices, and this is shown by the analysis by Barnhill et al [BARNHILL et al 88]. However, Van Wijk [VANWIJK 86] has established a solution using cubic Bezier patches when the number of adjacent points at each vertex is either three or four. Sarraga has established a solution involving Bezier patches of order six when the number of adjacent points to each vertex is either three, four or five [SARRAGA 87, SARRAGA 89]. These solutions show increased application over the rectangular problem, and offer scope for further development which will be exploited in Chapter 6. It is a considerable advantage that all patches in the surface are of the same type, mainly because only one calculation and one display algorithm are required.

### 5.7 Manipulative Ability of Surfaces

One of the key features for the implementation of surface modelling and computer aided design techniques in prosthetics and orthotics is the capability to be able to adjust or manipulate a surface to improve its characteristics. Therefore, in this section the manipulative capabilities of the surface techniques
presented in this chapter are discussed. There are three particular forms of manipulation considered, ranging from large scale manipulations where much of the surface is adjusted, through local, where a small localised area of the surface is changed, to 'micro' manipulations, where small scale adjustments are considered.

5.7.1 Large Scale Manipulations

The surface schemes discussed in detail in previous sections have been chosen to interpolate a given set of points \( \{p_i\} \). Typical large scale adjustments will desire the surface to interpolate a modified set of points \( \{p'_i\} \). After this adjustment of the interpolation points, the surface can be recalculated, and will still by its nature be interpolatory. Large scale manipulations are thus readily achieved for all the surface methods previously presented, and depend upon an understanding of how the \( \{p_i\} \) must be adjusted to achieve the desired result. Examples of large scale adjustments are the lengthening of an aircraft fuselage...
by increasing the scale along its length (and not by introducing a greater number of points \( \{ p_i \} \)), the lengthening of a lower limb socket in the UCL CASD program in Chapter 2, and the increase in width of a shoe last. An example of a large scale modification of a surface is given in Figure 5.33. Note that the relationship of the \( \{ p_i^* \} \) is the same as the relationship of the \( \{ p_i \} \). This relationship must be preserved if the routines for surface calculation are to be reused without modification.

5.7.2 Local Manipulations

Not all desired alterations and manipulations will be large scale. Some will be local, intended to affect only a small region of the surface. Examples of this might be the shape of the nose of an aeroplane, one of the rectifications to the socket shape in the UCL CASD system and extra relief in a shoe shape to allow for relief of pressure under one of the metatarsal heads. A local manipulation can be
achieved by the alteration of one or a few relevant points \( p_i \). Depending on the method of surface representation used, a different portion of the surface will be affected. To illustrate this, a rectangular problem will be considered in which the point \( p_{ij} \) is altered to \( p'_{ij} \), and we shall discuss how much of the surface is affected by this one change. Note first, however, that for the interpolatory schemes, the adjusted surface will still interpolate all the other points, but the tangents there and the shape of the surface between these points may be affected. An example of a local modification is given Figure 5.34.

**Hermite and Bezier Interpolation**

If tangent and twist information have to be calculated from the positions \( \{ p \} \), then the alteration of one point \( p_{ij} \) will affect the calculated values of tangent and twist at the points \( p_{i+\alpha,j+\beta}, \alpha, \beta \in \{-1,0,1\} \). Then if the patch between points \( p_{ij}, p_{i+1,j}, p_{i,j+1}, p_{i+1,j+1} \) is \( s_{ij}(u,v) \), the alteration of the one point \( p_{ij} \) will affect the sixteen patches \( s_{i+\gamma,j+\delta}, \gamma, \delta \in \{-2,-1,0,1\} \), Figure 5.34.

If the tangent and twist information does not have to be calculated, but is prescribed by some other criteria, then adjustment of one point \( p_{ij} \) will affect only the four patches \( s_{i+\gamma,j+\delta}, \gamma, \delta \in \{-1,0\} \).

**Non-Interpolatory B-Splines**

For a tensor product B-spline surface, where the points \( p_{ij} \) are used as control points, the points \( p'_{ij} \) are not interpolated, and this is the type of surface referred to here as 'non-interpolatory'. Recall from section 5.2.6 that one property of B-splines is their local control. Therefore for the cubic B-spline case where \( p_{ij} \) is associated with knot values \( u_i, v_j \), and the knot values remain unaltered, the alteration of \( p_{ij} \) affects the surface where \( u_{i-3} < u < u_i \) and \( v_{j-3} < v < v_j \). If the knot values are also affected, and become \( u^*_i, v^*_j \) such that the only the intervals affected are \( |u_i - u_{i-1}|, |u_{i+1} - u_i|, |v_j - v_{j-1}|, |v_{j+1} - v_j| \) then alteration of \( p_{ij} \) affects the portion of the surface where \( u_{i-4} < u < u_{i+1} \) and \( v_{j-4} < v < v_{j+1} \).
Interpolatory B-Splines

If the B-spline surface scheme used is such that the B-spline surface is forced to pass through the points \{p_j\}, then the calculation of the control points involves the inverse of a matrix of full rank. In other words, movement of one interpolation point \( p_j \) affects the position of all the control points for the B-spline surface, and therefore modifies the entire surface.

5.7.3 Micro Manipulations

In the section on local manipulations the effect of alteration of one of the data points upon the surface has been considered. However, it may be desirable to consider even more localised changes to the surface at points in between the data points \( \{ p_i \} \). Examples are the alteration required to the shape to fit a small component to an aircraft fuselage, local alteration of socket shape in the UCL CASD system to avoid a tender spot on the skin, or a minor aesthetic change to the shape of a shoe last. To evaluate the abilities of the previously presented surface schemes for micro manipulation, we consider what freedom there is other than repositioning of the points \( p_j \) in a rectangular case.

Hermite Interpolation

The only modifications available other than repositioning of the points \( p_j \) are adjustments of the tangents and twists at these points. However, as seen in the previous section, adjustments of these features affects all adjacent patches.

Bezier Interpolation

Bezier interpolation seems at first glance to offer only the same modifications as Hermite interpolation. This would be expected in view of the fact that they are essentially the same scheme. However, Bezier surfaces can be raised in degree very easily. If a cubic Bezier surface is raised one order to a quadric Bezier
surface, then the Bezier point \( b_{22}^* \) can be freely adjusted without affecting the GC\(^1\) properties of the surface, Figure 5.35. Similarly, if two adjacent patches are raised to quadric patches, then the three Bezier points shown in Figure 5.36 can be adjusted so that the GC\(^1\) properties remain. The simplest method of achieving this would be that if \( b_{20} \) were moved to \( b_{20}^* \), then \( b_{21}^{(1)}, b_{21}^{(2)} \) are moved to

\[
\begin{align*}
    b_{21}^{(1)} &= b_{21}^{(1)} + [b_{20}^* - b_{20}] \\
    b_{21}^{(2)} &= b_{21}^{(2)} + [b_{20}^* - b_{20}]
\end{align*}
\]

(5.66)

**Interpolatory and Non-Interpolatory B-Splines**

The only flexibility available to B-spline surfaces while keeping the surface a B-spline surface is to adjust the knot spacing, but any change will affect a region extending right across the surface. Greater flexibility has been achieved by Barsky [BARSKY 88, BARSKY & DEROSE 85] by generalising the concept of a
Figure 5.36 For two adjacent quartic Bezier patches the three indicated control points can be adjusted by the same amount without affecting the cross boundary tangent continuity.

B-spline to a Beta-Spline by the introduction of a tension parameter, but changes in the tension are global modifications. The Bezier micro adjustments of the previous paragraph can, of course, be realised if the surface is expressed in Bezier form.

5.7.4 Remarks

In this section we have concentrated upon the capability for manipulation of surface techniques previously discussed, especially in the case of a rectangular situation. The results for Bezier and Hermite interpolation carry over directly into the non-rectangular case. Bezier triangle interpolation has very similar manipulation properties to the tensor product Bezier patches already discussed. Coons patch properties and Shephard's methods properties have not been
discussed because they are not considered ripe for exploitation for use in the current prosthetics and orthotics project. Tensor product Bezier patch surfaces appear to offer the greatest flexibility, especially on a local and micro level.

5.8 Conclusions

In this chapter, common methods of representing surfaces using computer calculation have been reviewed. Since the problem in which these surface schemes will be implemented will be of an interpolatory nature, particular stress was put on the ability of the schemes to handle an interpolatory situation. A further feature of the surface scheme adopted will be its need to be flexible and to be able to represent a variety of adjustments and manipulations. This ability was also evaluated.

In the light of the discussion and examples given, the most appropriate technique for surface modelling and representation in the field of prosthetics and orthotics was thought to be a scheme using tensor-product Bezier patches with at least $GC^1$ continuity. However, the scheme will be used in a largely-rectangular situation, and no available algorithm has been found which satisfies the condition of general applicability. Some development will therefore be required, and this is the subject of the next chapter, together with derivation of simple and explicit formulae for the implementation of the scheme.
Chapter 6

A New Surface Model for General Application in Prosthetics and Orthotics

6.1 Introduction

6.2 Precise requirements of a surface model for prosthetics and orthotics

6.3 Development of a new surface model
   6.3.1 Vertices of order four
   6.3.2 Vertices orders three and five

6.4 Tangent estimation
   6.4.1 The tangent plane at a vertex of order three
   6.4.2 Tangent directions at a vertex of order four
   6.4.3 The tangent plane at a vertex of order five
   6.4.4 Tangent vectors at a vertex of order four
   6.4.5 Tangent vectors at vertices of orders three and five
   6.4.6 Tangent estimation from intermediate information

6.5 Twist Estimation
   6.5.1 Twist vectors at a vertex of order four
   6.5.2 Twist vectors at vertices of orders three and five
   6.5.3 Twist estimation from intermediate information

6.6 Formulae for the Bezier points of the new surface model
6.7 Algorithms for local modifications of the new surface model

6.8 Remarks
6 A New Surface Model

6.1 Introduction

In the previous chapter common methods of Computer Aided Geometric Design and surface modelling were reviewed. None was found to be ideal for general application to situations arising in prosthetics and orthotics, and therefore it was decided to develop a new surface model based upon tensor product Bezier patches in a nearly rectangular situation as suggested by sections 5.4.6 and 5.6.4. The new model will interpolate a number of specified points in three-dimensional space, and it will be suitable to many applications within prosthetics and orthotics.

Precise requirements on the specified points and the subsequent parametrisation are given in section 6.2, and the development of the new surface model itself is given in section 6.3. In the portions of the model which are non-rectangular, a new technique for tangent estimation is required, and this is developed in section 6.4, together with its extension to a new method for estimating tangents in the rectangular regions. Similarly, twist estimation is required, and a technique is presented in section 6.5. In section 6.6, the methods for estimating tangent and twist information and the generation of the surface model itself, are reduced to explicit expressions for positioning the Bezier points of the surface. Implementation will then be via calculation methods suggested in the previous chapter.

At the end of chapter 5, the manipulative ability of surfaces was discussed because of its practical importance in prosthetics and orthotics. Section 6.7 will give algorithms for local manipulation of the new surface model for those situations when the techniques of section 5.7.2 are invalid.

6.2 Precise Requirements of a Surface Model for Prosthetics and Orthotics

As suggested in the introduction to this chapter, the surface model sought will be a piecewise tensor product Bezier surface, or a 'Bezier patch' surface, applied
in a largely rectangular situation. In section 5.6.4 examples were shown of such largely rectangular situations (see Figure 5.32). Each point, \( p \), which the surface will be required to interpolate will be associated with a vertex of a mesh, where vertices of the mesh are connected to adjacent vertices by edges, and the edges enclose faces. Each face of the mesh is enclosed by exactly four edges to allow the respective portion of the surface to be represented by a tensor product Bezier patch. The requirements, each of which is subsequently discussed, of the surface model to be developed are:

- (1) The model will be applicable to any mesh which is largely rectangular with the exception of some non-adjacent vertices which can be of orders three or five such that the distances between points adjacent on the mesh are fairly similar.
- (2) The model will interpolate the given points \( p \), which are associated with the vertices of the mesh.
- (3) The model will comprise cubic or quartic tensor product Bezier patches over a uniform parametrisation.
- (4) The whole surface will be smooth.
- (5) The model will include a scheme for estimating tangent and twist vectors at the points for use when these vectors are not previously specified.

(1) Mesh Shape Figure 6.1 shows example meshes for various sections of the human body. Since these are a variety of shapes it is clear that a mesh with vertices of orders three, four and five will be widely adaptable and sufficient for many shapes encountered in prosthetics and orthotics. For completeness, however, a couple of examples of closed shapes, a sphere and an ellipsoid, are given in Figure 6.2. The vertices of order other than four are relatively few, and can be restricted so that no two are adjacent. The restriction on the allowable distances between adjacent points will be derived in discussion of requirement (3), the parametrisation.

(2) Interpolation Each tensor product Bezier patch would interpolate the four points associated with the four vertices at the corners of the face. This interpolation is readily achieved because of the properties of Bezier patches.
Figure 6.1 Example meshes for parts of the human anatomy. Vertices of orders other than 4 are indicated.

Figure 6.2 Example meshes for closed shapes - a sphere and an ellipsoid.
(3) Bezier patches and parametrisation

In a rectangular situation, cubic Bezier patches have enough freedom to interpolate the given points, but in a non-rectangular situation quartic patches will be required. Quartic patches will also give extra flexibility for manipulation of the surface in the rectangular regions. Section 4.5.4 showed that conventional methods within a rectangular problem require the ratio of the control points across the boundary of two adjacent patches to be the same (see Figure 5.25). In fact, this requirement is more severe in a rectangular region because it holds across each edge of each patch - the ratio across all the joins of adjacent rows of Bezier patches must be the same for the result to hold. In the non-rectangular cases under consideration, much of the mesh will be rectangular and suffer from this restriction. Figure 6.3 shows a possible boundary of two adjacent rows of patches in a largely-rectangular situation such as those under consideration. Since the row boundary spirals, it would seem that the most sensible ratio would be unity, otherwise the parametrisation becomes very complicated. A unity ratio requires the parametrisations of the patches to be the same for each Bezier patch, ie $0 \leq u \leq 1$ and $0 \leq v \leq 1$. Moreover, from the remarks of section 4.5.7 about the
difficulty of establishment and unreliability of a parametrisation, this does indeed seem the wisest choice. The only restriction is that the distances between adjacent vertices of the mesh must be fairly consistent to avoid strange shapes such as loops. If $K$ is the minimum distance between adjacent vertices, and $L$ is the maximum, a restriction is imposed such that for some $\kappa > 0$,

$$L < \kappa K$$  \hspace{1cm} (6.1)

A typical value of $\kappa$ might be 1.5, 2.0 or 3.0. In section 6.4.4, this condition and the value of $\kappa$ will be refined further to become $\sqrt{3}$ (6.42).

(4) Smoothness Smoothness will be required over the entire surface - within the patches, at their joins and at all vertices. The smoothness within patches is guaranteed by the properties of tensor product Bezier patches. There will be smoothness along the joins in rectangular portions of the mesh if the ratios of distances between control points across joins are equal, as discussed in the previous paragraph, and smoothness at a rectangular vertex will be guaranteed if the tangent plane at the vertex is uniquely defined. Near vertices of orders other than four, the surface model will have to ensure smoothness.

(5) Tangent and Twist Estimation The model will require a scheme for estimating tangent and twist vectors at the points $p_i$ for use when those vectors are not previously specified. This will often be the situation in prosthetics and orthotics, where the only surface information known is that the surface must pass smoothly through the points $p_i$.

Van Wijk has developed a surface of bicubic Bezier patches over non-rectangular meshes [VANWIJK 87]. However, the surface only allows vertices orders three and four. Section 4.6.4 showed that this would require exactly 8 vertices of order 3 for a closed object, and the examples given above show that this is not generally applicable. Moreover, the surface does not interpolate the initial points, and therefore it is rejected. Sarraga’s method [SARRAGA 87, SARRAGA 89] with techniques for tangent and twist estimation would fulfil all the conditions except (3) since it requires tensor product Bezier
patches of degree 6. This is more involved than necessary. Farin has hinted at an approach to develop the new model, but does not give a final result [FARIN 90]. New development is therefore required.

6.3 Development of a New Surface Model

This section develops the new surface model to fit the requirements of interpolation and smoothness discussed previously. In the rectangular portions of the mesh, techniques from chapter 5 are adopted and full derivation of the surface model is given in the non-rectangular portions of the surface. The section is mathematically involved, and the important results which are used to give explicit formulae for the Bezier points of the surface later are (6.14), (6.16), (6.23), (6.26), (6.27), (6.28) and (6.29).

6.3.1 Vertices Order Four

Chapter 5 has shown that at a vertex of order four, cubic or higher order Bezier patches can be fitted together to fulfil the conditions of section 6.2. The first intermediate Bezier point along the edge of each patch, and the twist Bezier point within each of the patches will be specified according to the tangent and twist vectors at the corners by (5.13), (5.46) and (5.48). Thus a bicubic patch surrounded by four vertices of order four will be completely defined. If higher order patches are required, they are most easily obtained by degree elevation of a completely defined bicubic patch. The cases that require further development are the vertices of orders three and five.

6.3.2 Vertices Orders Three and Five

The vertices of orders three and five are the only vertices which make the mesh for the surface model non-rectangular, and they are restricted so that each such vertex is adjacent only to vertices of order four. Note that two of these exceptional vertices can be at opposite corners of the same face or patch.
Examples of the vertices under consideration here are the exceptional vertices in Figure 5.32. The surface model about the exceptional vertices will be developed as far as is practical in the context of a vertex order $n$, and the two cases $n = 3$ and $n = 5$ will be specified when required. Conditions for positional continuity across the joins of the patches will first be generated, followed by conditions upon the tangents at the exceptional vertex. This will be followed by conditions upon the Bezier points on the edges of the patches, then the twist Bezier points about the exceptional corner, and finally conditions for the Bezier points internal to the patches.

Consider $n$ tensor product Bezier patches, $s^{(i)}(u_i,v_i)$, $1 \leq i \leq n$, each of degree $m$ (that is $m = 3$ for cubic) about a vertex, $p$. Throughout this section, when $i$ is used to indicate a patch number, the index will be cyclic, that is $n + 1 = 1$. The parameters, $u_i, v_i$, are explained by Figure 6.4. Let the Bezier points for $s^{(i)}(u_i,v_i)$ be $b^{(i)}_{jk}$, $0 \leq j \leq m, 0 \leq k \leq m$. For positional continuity across the patch boundaries, the requirements are
Figure 6.5 The Bezier points for a system of quartic patches around a vertex of order 5. The points undetermined by position and tangent continuity conditions with other patches are indicated by crosses.

$$ b_{00}^{(i)} = p \quad \text{for } 1 \leq i \leq n $$

and

$$ b_{0j}^{(i)} = b_{j0}^{(i+1)} = c_j^{(i)} \quad \text{say} $$

For tangent plane continuity, or GC¹, the requirements are first that there is a common tangent plane to all patches at node $p$, and second that there is tangent continuity across each boundary of adjacent patches. The first condition is already assured if a tangent plane for $p$ is used to determine each of the points $c_j^{(i)}$. The second condition will constitute the remainder of work in this section.

At this stage, note that if each of the Bezier patches were bicubic and the boundary curves had been calculated, then there would only be one unspecified Bezier point in each patch near the vertex in question, namely $b_{11}^{(i)}$. It will turn out that a higher order patch is required, and will be obtained by degree elevation of the cubic patch. The existing edge conditions will fix the points
A geometrical interpretation of (6.2) is that the indicated vectors are equal.

c_{m}^{(i)}, c_{m-1}^{(i)}, b_{1m}^{(i)}, b_{1,m-1}^{(i)}, b_{m1}^{(i)}, b_{m-1,1}^{(i)}.

In the quartic case of a five-patch corner, that will leave only those control points marked by crosses in Figure 6.5 free for setting. In particular, from the tangents and twists at the adjacent vertices order four,

\begin{align}
 b_{1m}^{(i+1)} - c_{m}^{(i)} + b_{1m}^{(i)} - c_{m}^{(i)} &= 0 \\
 b_{m-1,1}^{(i+1)} - c_{m-1}^{(i)} + b_{m-1,1}^{(i)} - c_{m-1}^{(i)} &= 0
\end{align}

(6.2)

A geometrical interpretation of these conditions is given in Figure 6.6. Note that these conditions hold both before and after any degree elevation.

The tangent plane at any point can be defined by two linearly independent tangential vectors to the surface. Two tangential vectors are \( \frac{\partial r}{\partial u} \) and \( \frac{\partial r}{\partial v} \). These will be linearly independent for all instances except where one of the interior
angles at a corner of the patch is not less than 180°. This is a situation that will not be permitted in the current project, but is not a serious restriction. The condition for tangent continuity is now that the following are coplanar,

\[
\frac{\partial s^{(i)}}{\partial u_i}(0,v_i), \frac{\partial s^{(i)}}{\partial v_i}(u_{i+1},0), \frac{\partial s^{(i+1)}}{\partial v_{i+1}}(u_{i+1},0)
\]  

(6.3)

Now put \( t = v_i = u_{i+1} \), and after observation that the second and third terms in (6.3) are the same, the requirement is that the following are coplanar

\[
\frac{\partial s^{(i)}}{\partial u_i}(0,t), \frac{\partial s^{(i)}}{\partial v_i}(0,t), \frac{\partial s^{(i+1)}}{\partial v_{i+1}}(t,0)
\]

Equivalently, there exist \( \alpha, \beta, \gamma \) such that

\[
\alpha \frac{\partial s^{(i)}}{\partial u_i}(0,t) + \beta \frac{\partial s^{(i+1)}}{\partial v_{i+1}}(t,0) = \gamma \frac{\partial s^{(i)}}{\partial t}(0,t)
\]  

(6.4)

\( \alpha, \beta, \gamma \) may vary with \( t \), giving \( \alpha(t), \beta(t), \gamma(t) \), but for symmetry set \( \alpha(t) = \beta(t) \). Strictly we should have \( \alpha_i(t), \beta_i(t), \gamma_i(t) \), but symmetry around the vertex removes the dependence on \( i \). The cross boundary continuity condition (6.4) thus becomes

\[
\alpha(t) \left[ \frac{\partial s^{(i)}}{\partial u_i}(0,t) + \frac{\partial s^{(i+1)}}{\partial v_{i+1}}(t,0) \right] = \gamma(t) \frac{\partial s^{(i)}}{\partial t}(0,t)
\]  

(6.5)

Writing the condition in terms of the Bezier points and Bezier polygons, this is

\[
\alpha(t) \sum_{j=0}^{n} B^{(m)}_j(t) \left[ b^{(i)}_j - b^{(i)}_{j_0} \right] + m \sum_{j=0}^{n} B^{(m)}_j(t) \left[ b^{(i+1)}_j - b^{(i+1)}_{j_0} \right] = \gamma(t) m \sum_{j=0}^{n-1} B^{(m-1)}_j(t) \left[ b^{(i)}_{j+1,0} - b^{(i)}_{j_0} \right]
\]

Dividing by \( m \), collecting terms on the left hand side and using the notation \( e^{(i)}_j \), this reads
\[ \alpha(t) \sum_{j=0}^{m-1} B_j^m(t) \left[ b_{ij}^{(i)} - c_j^{(i)} + b_{j+1}^{(i+1)} - c_{j+1}^{(i)} \right] = \gamma(t) \sum_{j=0}^{m-1} B_j^{m-1}(t) \left[ c_{j+1}^{(i)} - c_j^{(i)} \right] \quad (6.6) \]

From (6.2) the term in square brackets on the left hand side of (6.6) is zero for \( j = m - 1, j = m \). Therefore the condition is

\[ \alpha(t) \sum_{j=0}^{m-2} B_j^m(t) \left[ b_{ij}^{(i)} - c_j^{(i)} + b_{j+1}^{(i+1)} - c_{j+1}^{(i)} \right] = \gamma(t) \sum_{j=0}^{m-1} B_j^{m-1}(t) \left[ c_{j+1}^{(i)} - c_j^{(i)} \right] \quad (6.7) \]

Consider (6.7) at \( t = 0 \),

\[ \alpha(0) \left[ b_{10}^{(i)} - c_0^{(i)} + b_{01}^{(i+1)} - c_0^{(i)} \right] = \gamma(0) \left[ c_1^{(i)} - c_0^{(i)} \right] \]

or, rewriting in terms of \( p \),

\[ \alpha(0) \left[ c_1^{(i-1)} - p + c_1^{(i+1)} - p \right] = \gamma(0) \left[ c_1^{(i)} - p \right] \]

Since \( \alpha(0) \neq 0 \), otherwise the condition would not be sensible, let

\[ \lambda = \frac{\gamma(0)}{\alpha(0)} \]

and set \( q_i = c_1^{(i)} - p \). Then

\[ q_{i-1} + q_{i+1} = \lambda q_i \quad 1 \leq i \leq n \quad (6.8) \]

This can be expressed in matrix form,
Thus \( \lambda \) is an eigenvalue of the matrix, giving a ‘characteristic equation’ which involves a determinant. For \( n = 3 \) this is

\[
\begin{vmatrix}
-\lambda & 1 & 1 \\
1 & -\lambda & 1 \\
1 & 1 & -\lambda
\end{vmatrix} = 0
\]

or

\[
-(\lambda - 2)(\lambda + 1)^2 = 0 \quad (6.9)
\]

For \( n = 5 \),

\[
\begin{vmatrix}
-\lambda & 1 & 0 & 0 & 1 \\
1 & -\lambda & 1 & 0 & 0 \\
0 & 1 & -\lambda & 1 & 0 \\
0 & 0 & 1 & -\lambda & 1 \\
1 & 0 & 0 & 1 & -\lambda
\end{vmatrix} = 0
\]

or

\[
-(\lambda - 2)(\lambda^2 + \lambda - 1)^2 = 0 \quad (6.10)
\]

From the form of the matrix, \( \lambda = 2 \) will be a solution for any \( n \), corresponding to \( q_1 = q_2 = \ldots = q_n \). From a geometrical consideration, this is not the desired solution. A little thought will give the correct solution. If the \( q_i \) were of equal magnitude and at regular angular intervals of \( \frac{2\pi}{n} \), as shown in Figure 6.7, a solution would be anticipated. Thus

\[
\lambda = 2 \cos \left( \frac{2\pi}{n} \right)
\]

For \( n = 3 \),

\[
\lambda = 2 \cos \left( \frac{2\pi}{3} \right) = -1 \quad (6.11)
\]
Figure 6.7 If the vectors $q_i$ are of equal magnitude and at equal angles, then $q_i = \lambda(q_{i-1} + q_{i+1})$ is anticipated for some $\lambda$.

For $n = 5$,

$$\lambda = 2 \cos \left( \frac{2\pi}{5} \right) = \frac{1}{2} (\sqrt{5} - 1)$$

Note that for $n = 4$, the result would be $\lambda = 0$ which corresponds with the solution used for a four-patch node.

To satisfy the conditions (6.8), it is necessary to perturb the $q_i$.

For $n = 3$, $\lambda = -1$ is a repeated root, and the conditions (6.8) reduce to only one condition, namely

$$q_1 + q_2 + q_3 = 0$$

The least squares perturbation is then to change $q_i$ to $q_i^*$, where
Figure 6.8 The effect of modifying the Bezier points about a vertex of order 3 to satisfy the tangent conditions at the vertex.

\[ q_i^* = q_i - \frac{1}{3} (q_1 + q_2 + q_3) \]  

(6.13)

or, in terms of the Bezier points

\[ c_1^{(i)} = c_1^{(i)} + p - \frac{1}{3} (c_1^{(1)} + c_1^{(2)} + c_1^{(3)}) \]  

(6.14)

Examples of the effect on control points are given in Figure 6.8.

For \( n = 5 \), \( \lambda = \frac{1}{2}(\sqrt{5} - 1) \) is again a repeated root, and the conditions (6.8) reduce to three conditions, namely
Figure 6.9 The effect of modifying the Bezier points about a vertex of order 5 to satisfy the tangent conditions at the vertex.

\[ q_1 + q_3 = \lambda q_2 \]
\[ \lambda q_1 + \lambda q_2 + q_4 = 0 \]
\[ q_2 + q_5 = \lambda q_1 \]

The least squares perturbation is then to change \( q_i \) to \( q_i^* \), where

\[
q_i^* = \frac{2}{5} q_i + \frac{\lambda}{5} (q_{i-1} + q_{i+1}) - \frac{\lambda + 1}{5} (q_{i+2} + q_{i-2})
\] (6.15)

or, in terms of the Bezier points

\[
c_i^{(*)} = c_i^{(*)} + p - \frac{1}{5} \left( 3c_i^{(*)} - \lambda (c_i^{(i-1)} + c_i^{(i+1)}) + (\lambda + 1) (c_i^{(i-2)} + c_i^{(i+2)}) \right)
\] (6.16)

Examples of the effect on control points are given in Figure 6.9.
Consider now (6.7) at \( t = 1 \), that is,

\[
0 = \gamma(1)[c_m^{(i)} - c_m^{(i-1)}]
\]

Thus, since the term in square brackets is not zero,

\[
\gamma(1) = 0
\]  \hspace{1cm} (6.17)

Observe from (6.7) that \( \gamma(t) \) is of one order higher in \( t \) than \( \alpha(t) \). The simplest solution would be to set \( \alpha(t) \) constant, and \( \gamma(t) \) linear in \( t \). Then to satisfy (6.7) and (6.17),

\[
\alpha(t) = \alpha_0
\]

\[
\gamma(t) = \lambda \alpha_0 (1 - t)
\]

Now (6.7) becomes

\[
\alpha_0 \sum_{j=0}^{m-2} B_j^m(t) [b_j^{(i)} - c_j^{(i)} + b_{j+1}^{(i-1)} - c_j^{(i-1)}] = \lambda \alpha_0 (1 - t) \sum_{j=0}^{m-1} B_j^{m-1} [c_{j+1}^{(i)} - c_j^{(i)}]
\]

Equating coefficients in \( t^{m-1}(1 - t) \)

\[
0 = \lambda \alpha_0 [c_m^{(i)} - c_m^{(i-1)}]
\]

However, all the terms on the right are non-zero, and so this condition cannot be satisfied. Therefore we cannot assume \( \alpha(t) \) constant and \( \gamma(t) \) linear in \( t \). Now try \( \alpha(t) \) linear and \( \gamma(t) \) quadratic in \( t \). From (6.7) and (6.17),

\[
\alpha(t) = \alpha_0 (1 - t) + \alpha_1 t
\]

\[
\gamma(t) = \gamma_0 (1 - t)^2 + \gamma_1 t (1 - t) + \gamma_2 t^2
\]  \hspace{1cm} (6.18)

To satisfy previously derived conditions, and setting \( \alpha_0 = 1 \), since \( \alpha_0 \neq 0 \) and all the other values can be scaled accordingly, the formulae (6.18) become
\[ \alpha(t) = (1 - t) + \alpha_t t \]
\[ \gamma(t) = (1 - t)(\gamma_t + \lambda(1 - t)) \]

Therefore (6.7) becomes

\[ [(1 - t) + \alpha_t t] \sum_{j=0}^{m-2} B_j^m(t)L_j^i = (1 - t)[\gamma_t + \lambda(1 - t)] \sum_{j=0}^{m-1} B_j^{m-1}M_j^i \]  \hspace{1cm} (6.19) \]

where

\[ L_j^i = b_j^{(i)} - c_j^{(i)} + b_{j+1}^{(i+1)} - c_j^{(i)} \]
\[ M_j^i = c_j^{(i)} - c_j^{(i)} \]

Equating coefficients in \( t^m(1 - t) \)

\[ 0 = \gamma_t M_{m-1}^i = \gamma_t(c_m^{(i)} - c_{m-1}^{(i)}) \]

Thus \( \gamma_t = 0 \) and (6.7) becomes

\[ [(1 - t) + \alpha_t t] \sum_{j=0}^{m-2} B_j^m(t)L_j^i = \lambda(1 - t)^2 \sum_{j=0}^{m-1} B_j^{m-1}M_j^i \]

Equating coefficients in \( t^{m-1}(1 - t)^2, t^2(1 - t)^{m-1}, t(1 - t)^m \) respectively,

\[ \frac{1}{2} m(m - 1)L_{m-2}^i = \lambda M_{m-1}^i \]
\[ \frac{1}{2} m(m - 1)L_2^i + \alpha_t mL_0^i = \lambda \frac{1}{2}(m - 1)(m - 2)M_2^i \]  \hspace{1cm} (6.20) \]
\[ mL_1^i + \alpha_t L_0^i = \lambda(m - 1)M_1^i \]

For the case \( m = 3 \), the bicubic patch case,

\[ 3\alpha_t L_1^i = \lambda M_2^i \]
\[ 3L_1^i + \alpha_t L_0^i = 2\lambda M_1^i \]
Combining these equations, using the result \( L_0^i = \lambda M_1^i \) from (6.8), and rewriting in terms of Bezier points, this becomes

\[
[c_3^{(i)} - c_2^{(i)}] - 2\alpha_1[c_2^{(i)} - c_1^{(i)}] + \alpha_2^3[c_1^{(i)} - c_0^{(i)}] = 0
\]

All the points in this equation have been specified, and so it is not in general possible to find \( \alpha_1 \) to fulfil this condition. In particular the condition requires that the four Bezier points in the equation are coplanar, which is not necessarily true, and there is no freedom to enforce this condition. Therefore, the bicubic case will not generally give a solution.

Consider the case \( m = 4 \), the biquartic patch case. Then (6.20) become

\[
\begin{align*}
\alpha_1 6L_2^i &= \lambda M_3^i \\
6L_2^i + 4\alpha_1 L_1^i &= 3\lambda M_2^i \\
4L_1^i + \alpha_1 L_0^i &= 3\lambda M_1^i
\end{align*}
\]  

(6.21)

Note that these are the only conditions, since the \( r^i, t^i, (1 - t), (1 - t)^3 \) coefficients have been equated by the choices of \( \alpha(t), \gamma(t) \), and (6.21) can be rewritten as

\[
\begin{align*}
6\alpha_1 L_2^i &= \lambda M_3^i \\
4\alpha_1^2 L_1^i &= 3\lambda \alpha_1 M_2^i - \lambda M_3^i \\
\alpha_1^3 L_0^i &= 3\lambda \alpha_1^2 M_1^i - 3\lambda \alpha_1 M_2^i + \lambda M_3^i
\end{align*}
\]  

(6.22)

Again, using \( L_0^i = \lambda M_0^i \) and substituting in terms of the Bezier points, (6.22) give

\[
-\left[c_4^{(i)} - c_3^{(i)}\right] + 3\alpha_1\left[c_3^{(i)} - c_2^{(i)}\right] - 3\alpha_1^2\left[c_2^{(i)} - c_1^{(i)}\right] + \alpha_1^3\left[c_1^{(i)} - c_0^{(i)}\right] = 0
\]

There is no requirement as yet on \( c_3^{(i)} \), so it can be chosen to satisfy this condition. \( \alpha_1 = 1 \) is chosen because in this case, the condition is that the curve defined by the \( c^{(i)} \) is a (degree-elevated) cubic Bezier curve, since
\[ e_4^{(i)} - 4e_3^{(i)} + 6e_2^{(i)} - 4e_1^{(i)} + c_0^{(i)} = 0 \]  
\[ (6.23) \]

For this reason, and following the observation that the repositioning of \( c_1^{(i)} \) commutes with degree elevation, a suitable method of obtaining \( e_2^{(i)} \) is to reposition \( c_1^{(i)} \), and then to degree elevate the patches to biquartic.

In terms of Bezier points, (6.22) now reads,

\[ 4[b_{1,1}^{(i)} - c_1^{(i)} + b_{1,1}^{(i+1)} - c_1^{(i+1)}] = 3\lambda[c_3^{(i)} - c_2^{(i)}] - \lambda[c_4^{(i)} - c_3^{(i)}] \]

or

\[ \frac{1}{2} [b_{1,1}^{(i+1)} + b_{1,1}^{(i)}] = A_i \]  
\[ (6.24) \]

where

\[ A_i = c_1^{(i)} + \frac{1}{8} \lambda \{3[c_1^{(i)} - c_2^{(i)}] - [c_4^{(i)} - c_3^{(i)}] \} \]  
\[ (6.25) \]

There is a unique solution for the \( b_{1,1}^{(i)} \) for the cases \( n = 3,5 \). For \( n = 3 \),

\[ b_{1,1}^{(1)} = A_1 - A_2 + A_3 \]

\[ b_{1,1}^{(2)} = A_1 + A_2 - A_3 \]  
\[ (6.26) \]

\[ b_{1,1}^{(3)} = -A_1 + A_2 + A_3 \]

For \( n = 5 \),

\[ b_{1,1}^{(1)} = A_1 - A_2 + A_3 - A_4 + A_5 \]

\[ b_{1,1}^{(2)} = A_1 + A_2 - A_3 + A_4 - A_5 \]

\[ b_{1,1}^{(3)} = -A_1 + A_2 + A_3 - A_4 + A_5 \]  
\[ (6.27) \]

\[ b_{1,1}^{(4)} = A_1 - A_2 + A_3 + A_4 - A_5 \]

\[ b_{1,1}^{(5)} = -A_1 + A_2 - A_3 + A_4 + A_5 \]
Examples are given in Figure 6.10.

The final requirement from (6.22) is, in terms of Bezier points,

$$6[b_{1,2}^{(i)} - c_2^{(i)} + b_{2,1}^{(i+1)} - c_2^{(i+1)}] = \lambda [c_4^{(i)} - c_3^{(i)}]$$

One technique for determining the points $b_{1,2}^{(i)}, b_{2,1}^{(i+1)}$ is to set them as the points required to make $b_{1,2}^{(i)}, b_{2,1}^{(i+1)}$ the Bezier points to define a cubic curve, and then to perturb $b_{1,2}^{(i)}, b_{2,1}^{(i+1)}$ by the least squares modification to satisfy the condition. Thus

$$b_{2,1}^{(i+1)} = \frac{1}{6} \{ -b_{0,1}^{(i+1)} + 3b_{1,1}^{(i+1)} + 3b_{1,1}^{(i+1)} - b_{2,1}^{(i+1)} \}$$

$$b_{1,2}^{(i)} = \frac{1}{6} \{ -b_{1,0}^{(i)} + 3b_{1,1}^{(i)} + 3b_{1,3}^{(i)} - b_{1,4}^{(i)} \}$$

$$b_{2,1}^{(i+1)} = b_{2,1}^{(i)} + \left( c_2^{(i)} + \frac{1}{12} \lambda [c_4^{(i)} - c_3^{(i)}] - \frac{1}{2} [b_{1,2}^{(i)} + b_{2,1}^{(i+1)}] \right) \quad (6.28)$$

$$b_{1,2}^{(i)} = b_{1,2}^{(i)} + \left( c_2^{(i)} + \frac{1}{12} \lambda [c_4^{(i)} - c_3^{(i)}] - \frac{1}{2} [b_{1,2}^{(i)} + b_{2,1}^{(i+1)}] \right)$$

All the continuity conditions are now satisfied. Moreover, note that the set up has included the possibility that two opposite corners of the same patch be vertices of orders other than four, and none of the mathematics has excluded this situation. However, $b_{2,1}^{(i)}$ is free and has yet to be determined. A sensible approach seems to be to set this point as close as possible to a point that would be given by a bicubic patch, ensuring that this is the last of the Bezier points in the patch to be fixed. Thus
This section has developed a new surface model for the regions of a largely rectangular mesh near the exceptional vertices which satisfies the conditions laid out in section 6.2. A number of conditions which the control points must fulfil have been found. It was noted that many of the conditions could be fulfilled by bicubic patches, and since bicubic patches are uniquely determined by tangent and twist vectors at their corners, sections 6.4 and 6.5 will seek expressions for the estimation of these tangent and twist vectors. In section 6.6, formulae for determining the control points of the biquartic patches will be developed, and these formulae will ensure overall $C^1$ continuity by using the surface model presented in this section.
6.4 Tangent Estimation

Various techniques for tangent estimation along curves were presented in section 5.3.1, and for tangent estimation on surfaces in section 5.5.1. These latter techniques could be applied to vertices of order four of the surface as suggested in section 6.3.1. However, these methods are not applicable to vertices of other orders. Here in section 6.4.1 a method for estimating the tangent plane at a vertex of order three will be developed, and in section 6.4.2 it will be shown that this method can be applied to a vertex of order four, giving a new method for tangent direction estimation independent of the parametrisation. In section 6.4.3, the method for tangent plane estimation is extended to a vertex of order five.

The actual requirement is not, however, for tangent planes or directions, but rather for tangent vectors. Therefore, in section 6.4.4, the methods of section 6.4.2 are extended to give tangent vectors at vertices of order four which in fact relax one of the constraints on the surface model of section 6.2. In section 6.4.5 a procedure of obtaining tangent vectors from the tangent plane methods developed here is given. The restrictions upon tangent vectors imposed by the new surface model developed in this chapter are satisfied by the tangent estimates given. Finally it may be that some surface information is known between the vertices. A method for tangent estimation in such a situation is suggested in section 6.4.6

6.4.1 The Tangent Plane at a Vertex of Order Three

Consider the vertex of order three \( p_0 \) with adjacent points \( p_1, p_2, p_3 \). Then set

\[
q_1 = p_1 - p_0
\]
\[
q_2 = p_2 - p_0
\]
\[
q_3 = p_3 - p_0
\]

There are two immediate choices for an estimation of the tangent direction, \( n \) at
The first is the normal direction to the plane through $p_1, p_2, p_3$. The second is the normal direction to the sphere through $p_0, p_1, p_2, p_3$ at $p_0$. After some consideration, the first choice is rejected because the normal direction to the plane would take more emphasis from points at a large distance from $p_0$ than points close to $p_0$. The second choice would have the opposite effect, and so is preferred.

The following development shows that if $q_1, q_2, q_3$ are not coplanar then the normal direction to the sphere at $p_0$ is given by

$$n = \text{unit}(q_1^2(q_2 \times q_3) + q_2^2(q_3 \times q_1) + q_3^2(q_1 \times q_2))$$

(6.26)

where $\times$ is the vector cross product, and $q_i^2 = q_i \cdot q_i$. Note that if $q_1, q_2, q_3$ are coplanar, the correct result is given.

Let $c$ be the centre of the sphere through $p_0, p_1, p_2, p_3$. Then

$$(p_0 - c) \cdot (p_0 - c) = (p_1 - c) \cdot (p_1 - c)$$

Thus

$$(p_0 \cdot p_0) - 2(p_0 \cdot c) = (p_1 \cdot p_1) - 2(p_1 \cdot c)$$

$$\Leftrightarrow$$

$$(p_0 - p_1) \cdot \left[ \frac{1}{2} (p_0 + p_1) - c \right] = 0$$

Thus $q_1 = p_1 - p_0$ is perpendicular to $\frac{1}{2} (p_0 - p_1) - c$. However, $q_1 \times q_2, q_1 \times q_3$ are two linearly independent vectors perpendicular to $q_1$. Therefore, for some $\alpha, \beta$,

$$c = \frac{1}{2} (p_0 + p_1) + \frac{1}{2} \alpha(q_1 \times q_2) + \frac{1}{2} \beta(q_1 \times q_3)$$

(6.27)

Similarly for some $\gamma, \delta, \epsilon, \zeta$. 
\[ c = \frac{1}{2} (p_0 + p_2) + \frac{1}{2} \gamma (q_2 \times q_3) + \frac{1}{2} \delta (q_2 \times q_1) \quad (6.28) \]

\[ c = \frac{1}{2} (p_0 + p_3) + \frac{1}{2} \epsilon (q_3 \times q_1) + \frac{1}{2} \zeta (q_3 \times q_2) \quad (6.29) \]

Subtracting (6.28) from (6.27) and taking the dot product of the result with \( q_2 \) gives

\[ 0 = \frac{1}{2} (p_1 - p_2) \cdot q_2 + \frac{1}{2} \beta [(q_2 \times q_3) \cdot q_3] \]

\[ \Rightarrow \quad \beta T = (q_1 - q_2) \cdot q_2 \]

where \( T = [(q_1 \times q_3) \cdot q_3] \).

Subtracting (6.29) from (6.27) and taking the dot product of the result with \( q_3 \) gives

\[ 0 = \frac{1}{2} (p_1 - p_3) \cdot q_3 + \frac{1}{2} \alpha [(q_3 \times q_2) \cdot q_3] \]

\[ \Rightarrow \quad \alpha T = (q_3 - q_1) \cdot q_3 \]

Substituting for \( \alpha, \beta \) in (6.27),

\[ c = \frac{1}{2} (p_0 + p_1) + \frac{1}{2T} [(q_3 - q_1) \cdot q_3] (q_1 \times q_2) + \frac{1}{2T} [(q_1 - q_2) \cdot q_2] (q_1 \times q_3) \]

The normal is in the direction \( p_0 - c \) or \( c - p_0 \). Multiplying by \( 2T \) this direction is

\[ n = \text{unit} (T q_1 + q_3^2 (q_1 \times q_2) + q_2^2 (q_3 \times q_1) - (q_1 \cdot q_3) (q_1 \times q_2) - (q_1 \cdot q_2) (q_3 \times q_1)) \quad (6.30) \]

Consider now \( T q_1 \):
6 A New Surface Model

\[ Tq_1 = [(q_1 \times q_2) \cdot q_3]q_1 \]
\[ = [q_1 \cdot (q_2 \times q_3)]q_1 \]
\[ = [(q_2 \times q_3) \times q_1] \times q_1 + (q_1 \cdot q_2)(q_2 \times q_3) \quad \text{using } (a \cdot c)b = (a \times b) \times c + (b \cdot c)a \]
\[ = [(q_2 \cdot q_1)q_3 - (q_3 \cdot q_1)q_2] \times q_1 + q_1^2(q_2 \times q_3) \quad \text{using } (a \times b) \times c = (a \times c)b - (b \cdot c)a \]

Using this in (6.30) gives, as required,

\[ n = \text{unit}(q_1^2(q_2 \times q_3) + q_2^2(q_3 \times q_1) + q_3^2(q_1 \times q_2)) \]

This is symmetric in \( q_1, q_2, q_3 \) as would be expected. For small \( q_1 \), \( \text{ie } |q_1| \ll |q_2|, |q_3| \),

\[ n = \text{unit}(q_2^2(q_3 \times q_1) + q_3^2(q_1 \times q_2)) \]

and so the normal is perpendicular to \( q_1 \) as would be expected. If \( q_1 \) is large, \( \text{ie } |q_1| \gg |q_2|, |q_3| \),

\[ n = \text{unit}(q_1^2(q_2 \times q_3)) \]
\[ = \text{unit}(q_2 \times q_3) \]

and so \( q_1 \) is unimportant.

6.4.2 Tangent Directions at a Vertex of Order Four

Consider a vertex \( p_0 \) with adjacent vertices \( p_1, p_2, p_3, p_4 \) in that cyclic order. In section 5.5.1 the tangent directions were established from a curve through \( p_1, p_0, p_3 \) and \( p_2, p_0, p_4 \) respectively. It is logical to do the same here. A generalisation from the sphere in the previous section would be a circle through each of these sets of points. However, the circle through \( p_1, p_0, p_3 \) is the intersection of the spheres through \( p_1, p_0, p_3, p_2 \) and \( p_1, p_0, p_3, p_4 \). The normal directions to these spheres at \( p_0 \) say \( n_\omega, n_\beta \) will both be tangent to the required tangent, \( t \). Let
Then
\[ n_\alpha = (q_1^2(q_2 \times q_3) + q_2^2(q_3 \times q_1) + q_3^2(q_1 \times q_2)) \]
\[ n_\beta = (q_4^2(q_2 \times q_3) + q_3^2(q_3 \times q_1) + q_1^2(q_1 \times q_4)) \]

Consider
\[ t = q_3^2q_1 - q_1^2q_3 \]

The dot product with each of the normal vectors is
\[ t \cdot n_\alpha = q_3^2q_1^2((q_1 \times q_2) \cdot q_3) - q_3^2q_1^2((q_2 \times q_3) \cdot q_1) = 0 \]
\[ t \cdot n_\beta = q_4^2q_3^2((q_1 \times q_2) \cdot q_3) - q_4^2q_3^2((q_2 \times q_3) \cdot q_1) = 0 \]

Therefore the desired tangent direction is
\[ t = \text{unit}(q_3^2q_1 - q_1^2q_3) \] (6.31)

Note that if the point, \( p_0 \), is on the edge of the mesh, and if the points are relabelled as in Figure 6.11, all the above algebra holds, and (6.31) gives the tangent to the circle through \( p_1, p_0, p_3 \), which is a sensible estimate.

### 6.4.3 The Tangent Plane at a Vertex of Order Five

The first choice for the tangent plane at a vertex order five, \( p_0 \), with adjacent points \( p_1, p_2, p_3, p_4, p_5 \) would be an extension of the method for a vertex of order three with similar properties to those mentioned at the close of the subsection.
6.4.1. One approach would be to take the normal to the sphere which passes through \( p_0 \) and minimises the sum of the squares of the distances of \( p_1, p_2, p_3, p_4, p_5 \) to the surface of the sphere. However, the mathematics becomes very involved in this estimation, and so some direct extension of the formula in the vertex of order three case was sought.

Let

\[
q_1 = p_1 - p_0
\]
\[
q_2 = p_2 - p_0
\]
\[
q_3 = p_3 - p_0
\]
\[
q_4 = p_4 - p_0
\]
\[
q_5 = p_5 - p_0
\]

Then two vectors suggest themselves as possible normals.
\[ m = \text{unit}\left((q_1^2 + q_2^2 + q_3^2) (q_4 \times q_5) + \text{similar terms}\right) \]

\[ n = \text{unit}\left((q_1^2 q_2^2 q_3^2) (q_4 \times q_5) + \text{similar terms}\right) \]

The first of these, \( m \), is rejected because it does not have the required properties; when \( q_1 \) is large, the dependence on \( q_1 \) does not become insignificant, and when \( q_1 \) is small, \( m \) is not perpendicular to \( q_1 \). However, for the second choice, the required behaviour is apparent; when \( q_1 \) is small, \( n \) is perpendicular to \( q_1 \), and when \( q_1 \) is large, \( q_1 \) cancels out from the formula. The full expression for \( n \), which is the chosen vector, is

\[ n = \text{unit}\left((q_1^2 q_2^2 q_3^2) (q_4 \times q_5) + (q_1^2 q_2^2 q_3^2) (q_5 \times q_1) + (q_1^2 q_2^2 q_3^2) (q_2 \times q_3) + (q_1^2 q_2^2 q_3^2) (q_1 \times q_2) + (q_1^2 q_2^2 q_3^2) (q_3 \times q_1) + (q_1^2 q_2^2 q_3^2) (q_2 \times q_3) + (q_1^2 q_2^2 q_3^2) (q_1 \times q_2) \right) \]

\[ (6.32) \]

### 6.4.4 Tangent Lengths at a Vertex of Order Four

Here the tangent directions at a vertex of order four are used to generate tangent lengths and so tangent vectors. Consider the tangent at a vertex of order four, \( p_0 \), with adjacent vertices \( p_1, p_3 \) as described in section 6.4.2. Let

\[ q_1 = p_1 - p_0 \]

\[ q_3 = p_3 - p_0 \]

as in section 6.4.2. Then from (6.31) the unit vector in the tangent direction, \( t \) is

\[ t = \frac{1}{R} (q_2 q_1 - q_1 q_3) \]  

\[ (6.33) \]

where
Let the length of the tangent be $\alpha$. Then set $\lambda, \mu$ by

\[
\alpha(t \cdot q_3) = \lambda q_3^2 \quad (6.35)
\]
\[
\alpha(t \cdot q_1) = \mu q_1^2 \quad (6.36)
\]

That is, in words, that the projection of the tangent at $p_0$ in the direction of point $p_3$ is the fraction $\lambda$ of the vector $p_3 - p_0$. Figure 6.12. Similarly for $p_1$ and $\mu$. 

Figure 6.12 The projection of the tangent $t$ at $p_0$ in the direction of $p_3$ is the fraction $\lambda$ of the vector $p_3 - p_0$. 

\[
R = |q_3^2 q_1 - q_1^2 q_3|
\]
\[
= ((q_3^2 q_1 - q_1^2 q_3) \cdot (q_3^2 q_1 - q_1^2 q_3))^{1/2}
\]
\[
= (q_3^4 q_1^2 - 2q_3^2 q_1^2 (q_3 \cdot q_1) + q_1^2 q_3^4)^{1/2}
\]
\[
= (q_1^2 q_3^2 (q_3 - q_1) \cdot (q_3 - q_1))^{1/2}
\]
\[
= |q_1| \cdot |q_3| \cdot |q_3 - q_1| \quad (5.34)
\]
Substituting (6.33) in (6.35) and (6.36),

\[ \alpha = \frac{\lambda q_3^2 R}{(q_1^2 q_3 - q_3^2 q_1) \cdot q_3} = \frac{-\mu q_1^2 R}{(q_1^2 q_3 - q_3^2 q_1) \cdot q_1} \]

\[ \Rightarrow \quad \lambda((q_3 \cdot q_1) - q_3^2) = -\mu(q_1^2 - (q_1 \cdot q_3)) \]

If \( \theta \) is the angle between \( q_1 \) and \( q_3 \), and \( Q = \frac{|q_1|}{|q_3|} \), then

\[ \lambda(|q_3| |q_1| \cos \theta - |q_3|^2) = -\mu(|q_1|^2 - |q_1| |q_3| \cos \theta) \]

\[ \Rightarrow \quad \lambda = \frac{-1 + Q \cos \theta}{Q \cos \theta - Q^2} \]

\[ \mu = \frac{1 - Q \cos \theta}{Q(Q - \cos \theta)} \quad (6.37) \]

Suppose the surface is represented by cubic Bezier patches. Then the condition gained from (5.13) to avoid looped curves implies that the Bezier points corresponding to the edge \( p_0, p_3 \) should be non-decreasing in the direction \( q_3 \). A sufficient condition for a cubic Bezier edge is that the first intermediate Bezier point, \( b_1 \), should be such that \( b_1 - p_0 \) when projected onto \( q_3 \) should be less than \( \frac{1}{2} q_3 \) and more than \( rq_3 \) for some factor \( r, r < 1/2 \), Figure 6.13. Then, since

\[ b_1 - p_0 = \frac{1}{3} \alpha f \]

the condition is

\[ r|q_3| \leq \frac{1}{3} \alpha \left( f \cdot \frac{q_3}{|q_3|} \right) \leq \frac{1}{2} |q_3| \quad (6.38) \]
Figure 6.13 The projection of the vector $b_1 - p_0$ onto $q_3$ should be less than $\frac{1}{3}q_3$ and more than $r q_3$.

From (6.35)

\[ r q_3^2 \leq \frac{1}{3} \lambda q_3^2 \leq \frac{1}{2} q_3^2 \]

\[ \Leftrightarrow \]

\[ r \leq \frac{1}{3} \lambda \leq \frac{1}{2} \]

Recall that $0 < r < 1/2$. Similarly

\[ r \leq \frac{1}{3} \mu \leq \frac{1}{2} \]

Therefore

\[ 2r \leq \frac{\lambda}{\mu} \leq \frac{1}{2r} \]

In (6.37) this requires
\[ 2r \leq \frac{1 - Q \cos \theta}{Q(Q - \cos \theta)} \leq \frac{1}{2r} \quad (6.39) \]

For \( Q = 1 \), (6.39) holds for any \( \theta \). For \( Q > 1 \), (6.39) gives

\[ 2rQ^2 - 2rQ \cos \theta \leq 1 - Q \cos \theta \]

and

\[ 2r - 2rQ \cos \theta \leq Q^2 - Q \cos \theta \]

\[ \Rightarrow \]

\[ \cos \theta \leq \frac{1 - 2rQ^2}{Q(1 - 2r)} \]

and

\[ \cos \theta \leq \frac{Q^2 - 2r}{Q(1 - 2r)} \]

Sufficient conditions are that

\[ \cos \theta \leq 0 \]

\[ 2rQ^2 \leq 1 \]

\[ 2r \leq Q^2 \]

A similar argument can be followed for \( Q < 1 \). The result is

\[ \theta \geq \frac{\pi}{2} \quad (6.40) \]

\[ 2r \leq Q^2 \leq \frac{1}{2r} \quad (6.41) \]

The value of \( \alpha_{(q_3 \cdot q_3)} \) is restricted by (6.38). A desirable value for it to take would be \( \frac{1}{3} \) since from (5.14) this value with a similar tangent condition at \( p_3 \) would give a vanishing second derivative component in the \( q_3 \) direction. To make the range symmetrical about \( \frac{1}{3} \) this would require \( r = \frac{1}{6} \). That is
This can be guaranteed if (6.1) reads

\[ L \leq \sqrt{3} K \]  

(6.42)

However, this is more restrictive than necessary. The conditions (6.40) and (6.41) are required individually for each of the two pairs of 'opposite' vertices adjacent to each vertex of order four. However, neither condition is found to be too restrictive in practice.

All this algebra has not determined \( \alpha \), but only found sufficient conditions for \( \alpha \) to exist. It is determined by minimising the sums of the squares of the differences of the ratios \( \frac{\alpha (r \cdot q_i)}{3q_i^2} \) and \( \frac{\alpha (r \cdot q_i)}{3q_i^2} \) from 1/3. That is, minimise \( F \) with respect to \( \alpha \) where

\[
F = \left( \alpha \frac{(r \cdot q_3)}{3q_3^2} - \frac{1}{3} \right)^2 + \left( \alpha \frac{(r \cdot q_1)}{3q_1^2} - \frac{1}{3} \right)^2
\]

After substitution from (6.33) and some manipulation, this is equivalent to minimising \( G \) with respect to \( \alpha \), where

\[
G = \left[ \alpha (q_1^2 - (q_1 \cdot q_3)) - R \right]^2 + \left[ \alpha (q_3^2 - (q_1 \cdot q_3)) - R \right]^2
\]

\[
\frac{dG}{d\alpha} = 2(q_1^2 - (q_1 \cdot q_3)) [\alpha (q_1^2 - (q_1 \cdot q_3)) - R] + 2(q_3^2 - (q_1 \cdot q_3)) [\alpha (q_3^2 - (q_1 \cdot q_3)) - R]
\]

Setting this equal to zero, this requires

\[
\alpha \left[ (q_1^2 - (q_1 \cdot q_3))^2 + (q_3^2 - (q_1 \cdot q_3))^2 \right] = R [q_1^2 - 2(q_1 \cdot q_3) + q_3]
\]
It can be shown that

\[(q_1^2 - (q_1 \cdot q_3))^2 + (q_3^2 - (q_1 \cdot q_3))^2 = \frac{1}{2}((q_1^2 - q_3^2)^2 + |q_1 - q_3|^4)\]

Therefore, using (6.34), the condition for \(\alpha\) is

\[\alpha = \frac{2|q_1||q_3||q_1 - q_3|^3}{|q_1 - q_3|^4 + (q_1^2 - q_3^2)^2}\]

From (6.33), the tangent for the edge \(p_0p_3\) is

\[t = \frac{2|q_1 - q_3|^2}{|q_1 - q_3|^4 + (q_1^2 - q_3^2)^2}(q_3^2q_1 - q_2^2q_3) \quad (6.43)\]

For a vertex on the edge of the mesh, the minimisation is not so easy, and an alternative is to require \(\alpha = \mu\) in (6.36), giving

\[t = \frac{1}{q_3 \cdot (q_1 - q_1)}(q_3^2q_1 - q_2^2q_3) \quad (6.44)\]

### 6.4.5 Tangent Directions and Lengths at Vertices of Order Three and Five

The tangent direction, \(t\), at a vertex, \(p_0\), of order three or five corresponding to the edge \(p_0p_1\) where \(p_1\) is an adjacent vertex, is the direction from \(p_0\) to the projection of \(p_1\) into the tangent plane at \(p_0\). Figure 6.14, where the tangent plane is given by section 6.4.1 or 6.4.3. For \(q_i = p_1 - p_0\) that is

\[t = \frac{q_1 - (q_1 \cdot \hat{n})\hat{n}}{|q_1 - (q_1 \cdot \hat{n})\hat{n}|} = \frac{q_1 - (q_1 \cdot \hat{n})\hat{n}}{[q_1^2 - (\hat{n} \cdot q_1)^2]^{1/2}}\]
Piper suggests that the tangent of the cubic Bezier curve along the edge should be the length of the projection of \( \mathbf{q}_1 \) into the tangent plane plus one third the component of \( \mathbf{q}_1 \) in the normal direction [PIPER 87], that is for tangent length \( \beta \),

\[
\beta = |\mathbf{q} - (\mathbf{q} \cdot \mathbf{n})\mathbf{n}| + \frac{1}{3}(\mathbf{q} \cdot \mathbf{n})
\]

An alternative suggestion, equivalent to the argument used in section 6.4.4, is to set tangent length, \( \gamma \), so that the projection of the first intermediate Bezier point in a cubic Bezier curve along \( \mathbf{p}_0 \mathbf{p}_1 \) would be one third of the way from \( \mathbf{p}_0 \) to \( \mathbf{p}_1 \). This would require

\[
\gamma t \cdot q_1 = q_1^2
\]

If \( \phi \) is the angle between \( \mathbf{q}_1 \) and \( \mathbf{n} \), then the conditions for \( \beta, \gamma \) are
\[ \beta = |q_1| \left( \sin \phi + \frac{1}{3} \cos \phi \right) \]
\[ \gamma = |q_1| \frac{1}{\sin \phi} \]

When \( \sin \phi \) is small, \( \gamma \) becomes very large which is undesirable, yet at other values \( \gamma \) is preferable to \( \beta \) because of the vanishing component of the second derivative in the \( q_1 \) direction.

The value of \( \phi \) is now sought for which \( \beta = \gamma \), and then a suggestion is made for which of the possible tangent lengths should be chosen.

\[ \beta = \gamma \]
\[ \Leftrightarrow \quad \sin \phi + \frac{1}{3} \cos \phi = \frac{1}{\sin \phi} \]
\[ \Leftrightarrow \quad 3 \sin^2 \phi + \sin \phi \cos \phi - 3 = 0 \]
\[ \Leftrightarrow \quad 3 - 3 \cos^2 \phi + \sin \phi \cos \phi - 3 = 0 \]

Either \( \cos \phi = 0 \) or
\[ \sin \phi = 3 \cos \phi \]
\[ \Leftrightarrow \quad \sin^2 \phi = 9 \cos^2 \phi \]
\[ \Leftrightarrow \quad 1 = 10 \cos^2 \phi \]

Observing that
\[ \cos^2 \phi = \frac{(\hat{a} \cdot q_1)^2}{q_1^2} \]

then if \( \frac{(\hat{a} \cdot q_1)^2}{q_1^2} \geq \frac{1}{10} \) set tangent \( t \) by
Let the tangent vectors about a vertex of order three be $t_1, t_2, t_3$, and those about a vertex of order five $t_1, t_2, t_3, t_4, t_5$. Then the $t_i$ are related to $q_i$ in the notation of (6.8), (6.13) and (6.16) by $t_i = \frac{1}{n} q_i$. Therefore, to satisfy the conditions of the surface model developed in section 6.3, the $t_i$ should be adjusted to $t_i^*$ where for a vertex of order three,

$$t_i^* = t_i - \frac{1}{3} (t_1 + t_2 + t_3)$$

and for a vertex of order five,

$$t_i^* = \frac{2}{5} t_i + \frac{\lambda}{5} (t_{i-1} + t_{i+1}) - \frac{\lambda + 1}{5} (t_{i+2} + t_{i-2})$$

where $\lambda = \frac{1}{2} (\sqrt{5} - 1)$.

### 6.4.6 Tangent Estimation from Intermediate Information

It may be that the positions $p_i$ of the points to be interpolated by the surface are not the only information known about the shape of the surface. There may be positions on the surface which are known between the $p_i$, but the surface is not required to interpolate these other positions but only be as close as practical to them. Figure 6.15 indicates the positions of points which may be known relative to the mesh of the vertices. Here these intermediate points are exploited to give a tangent estimation.
Suppose there are $m$ rows of intermediate points between each vertex, where usually $m = 2$ or $3$. Then the tangents at the vertex can be estimated by the methods just presented, but using the first intermediate point along each edge of the mesh instead of the adjacent vertex positions. The final tangent vector is then multiplied by $m$ to give a correct estimate for the vertex.

6.5 Twist Estimation

Various techniques for twist estimation with rectangular surfaces were presented in section 5.5.1. In section 6.5.1 these techniques are applied to vertices of order four of the new surface model. However, the techniques are not applicable to vertices of other orders, and section 6.5.2 will show that the surface model in fact specifies the tangents at these vertices. The positions of the vertices may not be the only known information about the surface shape. Section 6.5.3 gives a method for twist estimation from intermediate information.
6.5.1 Twist Vectors at a Vertex of Order Four

Let $p$ be a vertex of order four with adjacent vertices $p_1, p_2, p_3, p_4$ in that cyclic order. Let $r_1, r_2, r_3, r_4$ be the vertices of the patches which have a corner at $p$ which are opposite to $p$ and such that the four patches have corners at the vertices \{p, r_i, p_1, p_2\}. Let the tangent vector at $p$ corresponding to the edge $pp_i$ be $t_i$, that at $p_i$ corresponding to $p, p_i$ be $t_i'$, that at $p$, corresponding to $p, r_i$ be $s_i$, and that at $p_i$, corresponding to $p, r_{i-1}$ be $s_i^\#$, Figure 6.16. Then, using (5.56), the twist, $w_i$, which is closest to Adini’s twist, at $p$ for the patch defined by \{p, p_1, r_i, p_{i+1}\} is given by

$$w_i = \frac{1}{4} [(s_i - s_i^\#) - (s_{i+1} - s_{i+1}^\#) + (s_{i+2} - s_{i+2}^\#) - (s_{i+3} - s_{i+3}^\#) - r_i + r_{i+1} - r_{i+2} + r_{i+3}]$$

(6.49)

In fact this is Adini’s twist except when one or more of the adjacent vertices, $p_i$, is of an order other than four. If $p$ is at the edge of the mesh as in Figure 6.17, then an alternative for twist estimation must be used. A similar idea is
Figure 6.17 The notation for estimating the twist at a vertex on the edge of a mesh.

\[ w_i = \frac{1}{2} [2s_1 - (s_2 - s'_2) - 2s'_3 - t'_3 + t_3 - r_1 + r_2 + p_1 - p_3] \]  \hspace{1cm} (6.50)

### 6.5.2 Twist Vectors at Vertices of Orders Three and Five

Let \( p, p_i, t, \ell^i \) be defined as in the previous section with \( i \) assuming the appropriate values. Using the notation of section 6.3 for a biquartic patch solution and equation (5.48), the twist, \( w_i \), at \( p \) for the patch with corners \( \{p, p_i, r_i, p_{i+1}\} \), is given by

\[ w_i = 16[b^{(i)}_{11} - (p + [c^{(i)}_1 - p] + [c^{(i+1)}_1 - p])] \]  \hspace{1cm} (6.51)

Other definitions in terms of the notation of section 6.3 are

\[ t_i = 4(c^{(i)}_1 - p) \]  \hspace{1cm} (6.52)
\[ p_i = c_4^{(i)} \]  \hfill (6.53)

\[ t_i' = 4(c_3^{(i)} - p_i) \]  \hfill (6.54)

Define

\[ D_i = 3(p_i - p) - (t_i - t_i') \]  \hfill (6.55)

The edges are cubic, and (6.23) gives

\[ 6c_2^{(i)} = -p + 4c_1^{(i)} + 4c_3^{(i)} - p_i \]
\[ = 3p + t_i + 3p_i + t_i' \]  \hfill (6.56)

Also, from (6.25),

\[ A_i = c_1^{(i)} + \frac{1}{8} \lambda [3(c_3^{(i)} - c_3^{(i)}) - (c_2^{(i)} - c_3^{(i)})] \]
\[ = \left( p + \frac{1}{4} t_i \right) + \frac{1}{16} \lambda [3p_i - 3p + t_i' - t_i] \]
\[ = \left( p + \frac{1}{4} t_i \right) + \frac{1}{16} \lambda D_i \]  \hfill (6.57)

where \( \lambda = -1 \) for a vertex of order three, and \( \lambda = \frac{1}{2} (\sqrt{5} - 1) \) for a vertex of order five. Then, for a vertex of order three, from (6.26),

\[ \theta_1^{(i)} = A_i + A_{i+1} - A_{i+2} \]

Using (6.51), (6.55) and (6.57), the twist \( w_1 \) for a vertex of order three becomes

\[ w_1 = 4t_1 + 4t_2 - D_1 - D_2 + D_3 \]  \hfill (6.58)

For a vertex of order five, from (6.27) with \( \lambda = \frac{1}{2} (\sqrt{5} - 1) \),
and from (6.51), (6.55) and (6.57), the value for the twist vector, \( w_1 \), is

\[
b^{(i)} = A_i + A_{i+1} - A_{i+2} + A_{i+3} - A_{i+4}
\]


6.5.3 Twist Estimation from Intermediate Information

It may be that the positions \( p_i \) of the points to be interpolated by the surface are not the only information known about the shape of the surface. As with estimation of tangent vectors discussed in section 6.4.6, there may be positions on the surface which are known between the \( p_i \), but the surface is not required to interpolate these other positions but only be as close as practical to them (see again Figure 6.15). With tangent estimation, if there were \( m \) rows of intermediate points between each vertex, then the tangent obtained by using these intermediate points had to be multiplied by a factor of \( m \). If in twist estimation the first intermediate point along each edge of the mesh and the first intermediate point within the face are used instead of the adjacent vertex positions, then the twist vector obtained requires multiplication by a factor \( m^2 \) to give a correct estimate for the value at the vertex.

6.6 Formulae for the Bezier Points of the New Surface Model

After the tangent and twist values have been determined at the corners of a the patch the positions of the Bezier points can be (uniquely) determined for a bicubic Bezier patch using (5.43), (5.13) and (5.48). From (5.15) these points can be used to determine the Bezier points of a biquartic patch after degree elevation. This derivation together with the necessary perturbation of the control points to ensure \( G^1 \) continuity are given in this section. Typically, the position of one Bezier point of each 'type' (corner, edge and interior Bezier points, for example)
Figure 6.18 The relationship between tangent and twist vectors and Bezier points for establishing explicit formulae for the points.

will be given, and the word 'similarly' will indicate that other Bezier points of the same type can be determined using the same formulae and construction with the vertices relabelled appropriately.

Let $p, p_1, p_2, t_1, t_2, t', t'', w_i$ be three position, four tangent and one twist vectors respectively, as defined in section 6.4, and let them be related to the layout of the Bezier points, $b_j, 0 \leq i, j \leq 4$, as shown in Figure 6.18. Then, using (5.43), the corner point, $b_{00}$, is given by

$$b_{00} = p \quad (6.60)$$

Similarly for $b_{04}, b_{40}, b_{44}$. Using (5.13), the first Bezier point, $b_{10}$, along the edge $pp_i$, is given by

$$b_{10} = p + \frac{1}{4} t_i \quad (6.61)$$
Similarly for \( b_{03}, b_{01}, b_{30}, b_{41}, b_{43}, b_{14}, b_{34} \). Since each of the edges is actually cubic, the mid Bezier point of each edge can be calculated using (5.15). For \( b_{20} \) this gives

\[
b_{20} = -\frac{1}{6} b_{00} + \frac{2}{3} b_{10} + \frac{2}{3} b_{30} - \frac{1}{6} b_{40} \quad (6.62)
\]

Similarly for \( b_{02}, b_{42}, b_{24} \). The interior Bezier point nearest each corner is determined by the twist vector at that corner from (5.48). Therefore

\[
b_{11} = b_{00} + [b_{01} - b_{00}] + [b_{10} - b_{00}] + \frac{1}{16} w_1 \quad (6.63)
\]

Similarly for \( b_{13}, b_{31}, b_{33} \). For a degree-raised bicubic patch, the interior point \( b_{21} \) is also determined using (5.15), that is

\[
b_{21} = -\frac{1}{6} b_{01} + \frac{2}{3} b_{11} + \frac{2}{3} b_{31} - \frac{1}{6} b_{41} \quad (6.64)
\]

Similarly for \( b_{21}, b_{32}, b_{23} \). This leaves only one point to be determined, \( b_{22} \). Again, (5.15) can be used either on the points \( b_{12} \) or on the points \( b_{21} \). For a degree-raised bicubic patch, these will give the same result. Alternatively, both sets of points can be used, and this will be advantageous after perturbations have been considered. That is,

\[
b_{22} = -\frac{1}{12} b_{02} + \frac{1}{3} b_{12} + \frac{1}{3} b_{32} - \frac{1}{12} b_{42} - \frac{1}{12} b_{20} + \frac{1}{3} b_{21} + \frac{1}{3} b_{23} - \frac{1}{12} b_{24} \quad (6.65)
\]

All the Bezier points have now been determined, and thus the surface is determined. When the patch is a degree-raised bicubic patch, that is when each of the corner vertices is of order four, \( GC^1 \) continuity will be ensured. However, if vertex \( p \), say, is of order three or five, then points \( b_{12}, b_{21} \) will have to be modified by correction factors according to (6.28).

Returning to the notation of section 6.3, consider the correction factor, \( F_i \), along edge \( i \) from point \( p \). That is the correction factor for \( b_{12}^{(i)}, b_{21}^{(i+1)} \). Then, from (6.28)
\[ F_i = c_2^{(i)} + \frac{1}{12} \lambda [c_4^{(i)} - c_3^{(i)}] + \frac{1}{2} [b_{12}^{(i)} + b_{21}^{(i+1)}] \]  

(6.66)

Now, from (6.54),

\[ c_4^{(i)} - c_3^{(i)} = -\frac{1}{4} l_i \]

and from (6.64),

\[ \frac{1}{2} [b_{12}^{(i)} + b_{21}^{(i+1)}] = \frac{1}{2} \left( \frac{1}{6} b_{10}^{(i)} + \frac{2}{3} b_{11}^{(i)} + \frac{1}{6} b_{14}^{(i)} - \frac{1}{6} b_{21}^{(i+1)} + \frac{2}{3} b_{11}^{(i+1)} + \frac{2}{3} b_{14}^{(i+1)} - \frac{1}{6} b_{21}^{(i+1)} \right) \]

\[ = -\frac{1}{12} [b_{10}^{(i)} + b_{01}^{(i+1)}] + \frac{2}{3} \frac{1}{2} [b_{11}^{(i)} + b_{11}^{(i+1)}] + \frac{2}{3} \frac{1}{2} [b_{13}^{(i)} + b_{31}^{(i+1)}] - \frac{1}{6} \cdot \frac{1}{2} [b_{14}^{(i)} + b_{41}^{(i+1)}] \]

Since the vertices adjacent to \( p \) are all of order four,

\[ \frac{1}{2} [b_{14}^{(i)} + b_{41}^{(i+1)}] = p_i \]

\[ \frac{1}{2} [b_{13}^{(i)} + b_{31}^{(i+1)}] = c_3^{(i)} \]

From (6.24),

\[ \frac{1}{2} [b_{11}^{(i)} + b_{11}^{(i+1)}] = c_1^{(i)} + \frac{1}{8} \lambda [3(c_3^{(i)} - c_2^{(i)}) - (c_4^{(i)} - c_3^{(i)})] \]

\[ = c_1^{(i)} + \frac{1}{16} \lambda [3p^i - 3p + l_i - t_i] \]

From (6.8),
\[ b_{i0}^{(j)} + b_{01}^{(j+1)} = [c_{i}^{(j-1)} - p] + [c_{1}^{(j+1)} - p] + 2p \]
\[ = \lambda [c_{i}^{(j)} - p] + 2p \]
\[ = \frac{1}{4} \lambda t_i + 2p \]

Using
\[ c_{2}^{(i)} = -\frac{1}{6} p + \frac{2}{3} c_{1}^{(i)} + \frac{2}{3} c_{3}^{(i)} - \frac{1}{6} p_i \]

the formula for the correction factor in (6.66) is
\[ F_i = \frac{1}{16} \lambda (t_i - t_i^* - 2[p_i - p]) \quad (6.67) \]

Notice that this vanishes if the edge is a degree-elevated quadratic edge. The definition of \( b_{12} \) now should read
\[ b_{12} = \left[ -\frac{1}{6} b_{10} + \frac{2}{3} b_{11} + \frac{2}{3} b_{13} - \frac{1}{6} b_{14} \right] + \frac{1}{8} \lambda [-b_{00} + 2b_{01} - 2b_{03} + b_{04}] \quad (6.68) \]

where \( \lambda = -1 \) if \( p \) is a vertex of order 3, and \( \lambda = \frac{1}{2} (\sqrt{5} - 1) \) if \( p \) is a vertex of order 6. If this definition is used for \( b_{12} \) and a similar definition for \( b_{21} \), then the definition for \( b_{22} \) can stand, and the Bezier surface will be completely defined and \( GC^1 \) overall.
6.7 Algorithms for Local Modification of the Surface

This chapter has demonstrated how a smooth surface can be generated which will interpolate a given set of data points \( \{p_i\} \). It often occurs that the surface is not quite the surface which was required, and requires local modifications. In this section three distinct techniques for local modifications will be examined, each of which will affect a different proportion of the surface. Large scale modifications are achieved by repositioning of the points \( p_i \), in exactly the same way as in the previous chapter.

Movement of a Data Point

The first technique for local modification of the surface is to move one of the data points. The point \( p_i \), could be moved by a distance \( \rho \) in the outward normal direction to the tangent plane at that point. This would affect each of the patches which had \( p_i \) as a corner. Moreover, if the movement of \( p_i \) were allowed to affect the tangent and twist vectors of adjacent points, the movement would affect all of the patches which had one or more points adjacent to \( p_i \), as corners. For minor modifications of a local nature, it is suggested to move \( p_i \) in the normal direction by amount \( \rho \), and not to let this affect the tangent and twist vectors at the corners. This will allow increased speed in modification of the surface.

Modification of the Central Control Point of a Bezier Patch

It can be desirable to alter the shape of a surface between the data points and while leaving the data points unaltered. It is clear from (5.46) that the modification of the central control point, \( b_{22} \), of a biquartic Bezier patch will not affect the GC\(^1\) continuity of the surface at the joins of the patches, see Figure 5.35. The point can therefore be moved at will. The only portion of the surface affected is the one Bezier patch within which the Bezier point is moved. If \( p_1, p_2, p_3, p_4 \) are the corner points of the Bezier patch in that cyclic order, and in a
clockwise sense when viewed from outside the surface, then \((p_2 - p_4) \times (p_1 - p_3)\) is a vector out of the surface and a suggested modification is movement of the point \(b_{22}\) by

\[
\beta(b_2 - b_4) \times (b_1 - b_3)
\]  
(6.69)

**Modification of the Mid-Point of an Edge of a Bezier Patch**

The movement of the central control point of a Bezier patch allows modifications to the surface within individual Bezier patches. However, for completeness it is also necessary to have some method of modification of two adjacent Bezier patches without affecting the data points. Consider a vertex \(p\) with adjacent vertices \(p_1, ..., p_n\) in clockwise order and Bezier points \(b_{i}^{(i)}, c_{j}^{(i)}\) as defined in section 6.3. If \(p\) and \(p_i\) are vertices of order four (and so \(n = 4\)) then if \(b_{12}^{(i)}, c_{2}^{(i)}, b_{21}^{(i+1)}\) are moved by the same amount, say \(d\), the \(G^1\) continuity will not be affected. This is the same as the situation in Figure 5.36. A suggestion for the movement, \(d\) is

\[
d = \gamma(b_{21}^{(i+1)} - b_{12}^{(i)}) \times (p_i - p)
\]  
(6.70)

However, if the vertex \(p\) is of order 3 or 5, then the situation is more complicated. From (6.23),

\[
4c_{1}^{(i)} = c_{0}^{(i)} + 6c_{2}^{(i)} - 4c_{3}^{(i)} + c_{4}^{(i)}
\]

Therefore, if \(c_{2}^{(i)}\) is altered by \(d\), then \(c_{1}^{(i)}\) must be altered by \(\frac{3}{2} d\). However, \(c_{1}^{(i)}\) must lie in the tangent plane at \(p\), and so in general this condition cannot be met. Therefore, this type of modification is not possible near vertices of orders 3 and 5. Modifications can most easily be affected there by alteration of the length of the tangent vectors at \(p\) although this will require recalculation of the whole solution about \(p\). For simplicity, it is recommended that modification is not permitted in this way near a vertex of order 3 or 5. If the vertices of orders 3 and 5 are rare within the model, then this is not a serious restriction, especially since the previous two types of modification are available.
6.8 Remarks

Based on the common methods of surface modelling and Computer Aided Geometric Design presented in the previous chapter, this chapter has developed a new surface model more generally applicable in prosthetic and orthotic applications. Conditions for the model to be appropriate have been established together with techniques for estimation of the factors required by the surface model. These include new methods for tangent estimation at the vertices of the mesh. Finally, consideration has been given to the ability to manipulate the surface by the use of local modifications, and methods for the modifications have been suggested.
Chapter 7

Adaption of the New Surface Model to Particular Situations and the Fitting of Data to it.

7.1 Introduction

7.2 Adaption of the new surface model to particular situations
   7.2.1 The prosthetic socket
   7.2.2 The whole foot
   7.2.3 The orthotic insole

7.3 Marker points

7.4 Fitting data to the new surface model
   7.4.1 Structured data files
   7.4.2 Dividing up the data and averaging
   7.4.3 Fitting data by curve fitting techniques

7.5 Remarks
7 Adaption and Fitting

7.1 Introduction

One of the reasons why the UCL CASD system documented in Chapter 2 of this thesis and in [REYNOLDS 88, REYNOLDS & LORD ip] was regarded as a first generation system was that it is limited in the situations to which it could apply. Chapter 6 has developed a new surface model which has been shown to be able to assume a large variety of shapes. However, it has not been shown how to adapt the surface model to a particular situation, nor how to fit the captured data file to the model - a process which will often involve the reduction of thousands of data points to hundreds of Bezier points. This chapter will tackle these two aspects of the implementation of the new surface model. ‘Application’ will be used to refer to the alteration of the layout of the new surface model for a specific purpose, and ‘Fitting’ will be used to refer to the fitting of a particular data set to the model in that specific application.

Section 7.2 will describe how to adapt the model to particular situations giving examples from prosthetics and orthotics. Section 7.3 will consider marker points, as suggested in the presentation of the three dimensional data editor of chapter 4, and their use in the context of fitting data to the model. Section 7.4 will present various techniques for fitting the captured data to the new surface model using the examples of section 7.2, and section 7.5 will conclude the chapter with remarks on the presented techniques for adapting and fitting the model.

7.2 Adaption of the New Surface Model to Particular Situations

Recall from section 6.2 that the new surface model is appropriate to largely rectangular situations with exceptional non-adjacent vertices of orders 3 and 5. This leads to a three stage procedure for adapting the model to a particular situation:-

(i) identification of the rectangular regions of the situation;
(ii) verification that the exceptional vertices required are of the allowed orders;
(iii) ensurance that the exceptional vertices are not adjacent.
Identification of the rectangular regions

The first stage of the process is to identify those parts of the situation which can be modelled by rectangular regions. Section 5.6.4 showed that there could be three types of rectangular regions: sheet, cylinder and torus. The latter is not very practical in prosthetics and orthotics, but the former two are common. As a rule of thumb, rectangular sheet regions will represent areas that are approximately flat, or can be visualised by bending a piece of paper to assume a similar shape without creating any sharp creases. Examples are the sole of the foot, the palm of the hand and the face. Although the regions are 'rectangular' in form, this does not require four sharp corners because the layout of the rectangular mesh can be affected so that it is rounded at its extremes, Figure 7.1. Rectangular cylindrical regions are common in prosthetics and orthotics, and are readily applied to legs, arms and fingers, for example. They can also be
A vertex of order 6 can be replaced by two vertices of order 5. Similar procedures can be adopted for vertices of higher order.

applied in less obvious places, such as the region of a foot which is covered by the upper of a shoe. The rectangular regions should cover the whole of the surface and not overlap each other.

(ii) Verification of the orders of the exceptional vertices
The exceptional vertices of the constructed mesh will all be where the rectangular regions join. The orders of these vertices will usually be three, four or five, and all vertices of these orders are readily accommodated within the new surface model. For a vertex to have order greater than five, it would have to be at the join of at least three rectangular regions, such as at the join of the two legs and the torso on the human body. However, such points are rare on the body, and the problem can be removed by altering slightly the layout of the mesh. Figure 7.2 shows an example of a vertex of order six which becomes two vertices of order five by such an alteration. To check that the vertices are of the correct order,
section 5.6.4 showed that if the mesh for the object is closed - that is if it completely contains a portion of space - then the number of vertices of order 5, \( s \) say, and the number of order 3, \( r \) say, are related by

\[
r = s + 8
\] (7.1)

If there are \( p \) open ends to the mesh, where an open end is apparent on the mesh as a ring of external vertices, and the open ends are surrounded by external vertices of order three (each external vertex being connected to two other external vertices and one internal vertex), with the exception of a total of \( q \) external vertices of order two (external vertices connected to two other external vertices and no internal vertices), and \( r, s \) refer only to internal vertices, then the relationship will be

\[
r = s + 8 - 4p - q
\] (7.2)
Figure 7.4 A cap can be added to the end of a cylindrical mesh to give a mesh suitable for a prosthetic socket including its distal end. The exceptional vertices are indicated.

As examples, consider the open ended rectangular cylinder, where $p = 2, q = 0, r = s = 0$, or the sphere of Figure 6.2, where $p = q = 0, r = 8, s = 0$, or a rectangular sheet, where $p = 1, q = 4, r = s = 0$.

(iii) Ensurance of non-adjacency of the exceptional vertices
If two exceptional vertices are adjacent, that is they lie on adjacent and not opposite corners of a face of the mesh, then they must be separated before application of the new surface model. This is most easily achieved by the introduction of a new line of the mesh between the two vertices, as shown in the example of Figure 7.3.
7.2.1 The Prosthetic Socket

It is simple to construct a mesh for a prosthetic socket since the area of concern is cylindrical in shape, and this can be modelled by one rectangular region, see Figure 2.11. In the UCL CASD system, this is the mesh used, and, at the distal end, the cylindrical shape gets smaller until it is finally cut off by a flat plane at its extreme. Using the new surface model it is possible to add a spherical cap shape to the end of the cylinder, thereby giving a smooth surface over the entire shape, Figure 7.4. The spherical cap shape is a second rectangular region. Note that all the exceptional points lie on its boundary. In this example, in the notation of equation (7.2), \( p = 1, q = 0, r = 4, s = 0 \).
Figure 7.6 A second possible mesh for a whole foot, corresponding to 'sole' and 'upper' shape. The rectangular regions and exceptional vertices are indicated.

7.2.2 The Whole Foot

There are two immediate philosophies for adapting the new surface model to represent the whole foot. The first is to visualise the foot as essentially a 'sock' shaped cylinder with a smooth cap on the toe end. The resulting mesh is then similar to the capped prosthetic socket model of the previous paragraph. The effect is that of putting the centre point of the mesh of Figure 7.5 over the toes and wrapping the rest of the mesh around the foot and up the lower leg. The rectangular regions are indicated in the figure, where again \( p = 1, q = 0, r = 4, s = 0 \).

The second philosophy is to treat the foot as two rectangular regions, and this may be particularly useful for shoe design applications. The first region is the sole of the foot, a rectangular sheet, and the second is the 'upper' of the foot, which is a modified cylinder with the axis of the cylinder roughly vertical in a
normal standing position. The mesh is shown in Figure 7.6 with the rectangular regions identified. In the notation of (7.2) the mesh is the same as that of the previous paragraph. This is indeed very similar to the mesh suggested for design of shoe uppers and employed by Clarks, for example [GUNDILL & HACKNEY 88].

### 7.2.3 The Orthotic Insole

The orthotic insole covers that portion of the sole of the foot proximal of the metatarsal heads, but including the sides of the foot up to a certain height. The sole can be considered as a rectangular region, and the sides of the foot another, as shown in Figure 7.7. There are two internal vertices of order three, and the rectangular regions are shown. In the notation of (7.2), $p = 1$, $q = 2$, $r = 2$, $s = 0$. 

![Figure 7.7 A mesh for the orthotic insole with rectangular regions, exceptional vertices and size parameters indicated.](image)
This mesh has the strong advantage that the vertices of exceptional orders are placed in highly curved regions for which they are best suited because the internal angles of the patches which meet there can almost be right angles.

### 7.3 Marker Points

In the presentation of the data file editor, VIEW3D, of chapter 4, the concept of marker points was introduced. The data file editor allows addition and movement of the data points graphically so that they can be freely used. In fact, UCL CASD uses one marker point, the 'reference point', to keep track of the orientation and alignment of the data files. It is difficult to match similar two-dimensional, let alone three-dimensional, shapes to one another without the aid of information like marker points to determine their respective alignment. Mokhtarian and MacWorth have developed a technique for matching curves by using curvature information, and Besl and Jain have a similar technique for surfaces which uses the principal curvatures at any point to characterise the surface there [MOKHTARIAN & MACWORTH 86, BESL & JAIN 86, BESL & JAIN 88], but such techniques are computer intensive and much more involved than using marker points, if they are available. The three dimensional editor is a straight-forward method of adding marker points to a data file, and if they are to be used to mark bony prominences or other features, the user should be able to determine the positions required from the pattern of the data points in that area. A method of automatically detecting a marker position for the vertebra prominens using local surface curvature has been suggested by Drerup and Hierholzer, but such a method is restricted to a situation where the behaviour of the surface curvature is known [DRERUP & HIERHOLZER 85].

Marker points can be advantageous in other respects too, such as when development of the captured data is required. Foulston et al, for example, have used marker points to identify which portion of their data file was relevant to the determination of slopes along the plantar surface of the foot [FOULSTON et al 90]. In the fitting of data sets to the new surface model, marker points will occasionally be attached to specific points on the surface mesh, and at other times
they will be used for guidance in extracting particular shape information. Exactly where marker points are required will depend on the particular situation involved and the method employed to fit data to the new surface model. The data fitting techniques of the following section will give examples of the use of marker points which may be indicative of how marker points could be used in other situations.

7.4 Fitting Data to the New Surface Model

At this stage in the implementation of the new surface model, there is a mesh adapted to the particular situation in which the model will apply, and a data file which consists of a series of (x,y,z) points on the surface of the object. In the mind it is possible to visualise fitting the surface model to the data with ease. However, to translate this process into a rigorous procedure is fairly involved, and there are several possible methods. The information sought is in the form of the positions of the vertices of the mesh. The methods of fitting the data detailed here will be demonstrated with examples taken from the applications of section 7.2. Tests of each of the methods given here will be presented in the next chapter.

In sections 6.4.6 and 6.5.3, the possibility of using information intermediate to the positions of the vertices to enhance tangent and twist estimation was considered. This is equivalent to dividing each face of the mesh into $m \times m$ sub-faces for $m = 2$ or $3$, and estimating the position of the vertices of all the sub-faces. Each of the methods given here could be extended to use intermediate information, and this will be demonstrated in the implementation of the first example when tests are presented in the next chapter.

7.4.1 Structured Data Files

It is possible for the data captured to be exactly that required by the new surface model. This is the situation with the prosthetic socket model of section 7.2.1 when applied to the data captured by the UCL CASD system. In this case, the
Figure 7.8 An example mesh for a below-knee prosthetic socket constructed from data for the UCL CASD system.

tangent and twist vectors and then the Bezier points for the surface can be immediately calculated by the techniques of chapter 6. An example of a mesh for a below-knee prosthetic socket model achieved in this manner is given in Figure 7.8.

7.4.2 Dividing up the Data and Averaging

A second method for fitting data to the new surface model is to divide the data up into portions, with one portion for each of the mesh points required, and then to estimate each mesh point from the data points of the respective portion of the data file. To achieve this, the data file should be divided into non-overlapping, approximately equally sized portions, each containing some data. In this context, 'equally sized portions' means that the portions should be associated with regions approximately equal in area on the surface of the object. If a handful of the portions contain no data, then data may be added to the captured data file
using the data editor, estimating by eye where a data point would lie in the empty portion. Using cross-sections, this estimation can be accurate. If a significant number of the portions contains no data, then the mesh being sought is too fine for the given data file, and a coarser one with less vertices should be used.

Typically, a three stage procedure will be involved in the implementation of this method:

(i) add marker points to the data file to facilitate the portioning of the file;
(ii) divide up the data file into distinct portions;
(iii) estimate the mesh point for each portion.

Elaboration of the stages of such a procedure is now given in the case of a whole foot model using the mesh of Figure 7.5. Similar considerations will be relevant in other situations.
(i) Adding Markers
During editing of the data file, the following seven markers were added, see Figure 7.9:-

\( p_1, p_2 \) - markers indicating the direction of the lower leg;
\( p_3, p_4 \) - the central points of the medial and lateral malleoli;
\( p_5, p_6 \) - the medial and lateral prominences of the first and fifth metatarsal heads respectively;
\( p_7 \) - the tip of the big toe.

(ii) Dividing up the Data File
An axis composed of three linear sections was established, the three sections being from \( p_7 \) to \( \frac{1}{2} (p_5 + p_6) \), from \( \frac{1}{2} (p_5 + p_6) \) to \( \frac{1}{2} (p_3 + p_4) \), and from \( \frac{1}{2} (p_3 + p_4) \) proximally in the direction \( p_1 - p_2 \). As a guide to how many slices there should be about this axis, the number of slices desired between the metatarsal heads and the malleoli was indicated. A value of 7 was found to be optimal in that this was the maximum value with which there were not a significant number of the final portions of the data file containing no data points. The data was sliced up along the axis at equal intervals, up to a height of two slices proximal of the malleoli. If \( n_1 \) is the unit vector in the \( p_1 - p_2 \) direction, and \( n_2 \) in the \( \frac{1}{2} (p_5 + p_6) - p_7 \) direction, then the slicing was done normal to the axis except in the region between the metatarsal heads and the malleoli. Here, at a position \( \lambda \left[ \frac{1}{2} (p_5 + p_6) \right] + (1 - \lambda) \left[ \frac{1}{2} (p_3 + p_4) \right] \) on the axis, the slicing was done normal to the direction \( \lambda n_2 + (1 - \lambda) n_1 \). The result was slicing as shown in Figure 7.10.

Each slice of data was then divided up into twelve sectors at equal angular intervals, except for the most distal slice which was divided up into only six sectors because of the form of the surface model mesh in that region (Figure 7.5). The central point about which the angular segmentation was applied was on the axis except between the metatarsal heads and the malleoli, where the centre for segmentation in any one slice was estimated as the average of the \((x,y,z)\) extremes of the data points in that slice. The reason for this estimation was so that the central point would be closer to the centre of the data points in the slice.
The average position of the data points themselves was not considered an appropriate central point since the data points may not be evenly distributed around the slice.

(iii) Estimating the Mesh Points
The simplest method for estimation of the mesh points was to average the data points in each portion of the file. This also has the advantages of being swift and immediate. A second technique, that of choosing one data point from each portion to be representative of the data in that portion was also considered, and the iterative algorithm developed for determination of the point was

\[ r_i = \frac{1}{n} \sum_{j=1}^{n} p_j \]
The algorithm converges to one of the points except if there are only two points, when the algorithm yields the average position. This technique was eventually rejected because it gave no improvement over simple averaging and yet increased the calculation time required. An example of the final mesh produced by simple averaging is given in Figure 7.11.

7.4.3 **Fitting Data by Curve Fitting Techniques**

A third method for fitting the data to the new surface model in a particular situation is to use polynomial curve fitting techniques. This method involves projecting some of the data points to a two-dimensional plane and then using
polynomial curve fitting techniques for an ordinate variable against a dependent variable to generate a good polynomial approximation to the data points. Vertex positions are then read off the curve at regular intervals and their positions projected back into the original three dimensional data file. Suitable polynomial curves of best fit are developed in [BEVINGTON 69] and [FROBERG 81], for example. In a polynomial curve fitting method, a five stage procedure is typical:-

(i) add marker points to the data file to facilitate the processing of the file;
(ii) select relevant data points and project the situation into a suitable plane;
(iii) fit a polynomial curve to the data;
(iv) read positions off the curve at required intervals;
(v) project the positions back into the original three dimensions.

Elaboration of the stages of such a procedure is now given in the case of an orthotic insole model using the mesh of Figure 7.7. Similar considerations will be relevant in other applications.

(i) Adding Marker Points
The following marker points, several of which are suggested by Foulston et al [FOULSTON et al 90], were added to the data file, see Figure 7.12:-

$p_1, p_2, p_3$ - the bony prominences of the 1st, 2nd and 4th metatarsal heads on the plantar surface of the foot;
$p_4$ - a midfoot point at the cross between a line extended distally from a line bisecting the posterior calcaneus and a line dropped from the tuberoscity of the navicular;
$p_5$ - a midheel point on the plantar surface;
$p_6, p_7$ - the maximum height of the insole indicated by markers medially and laterally opposite the first and fourth metatarsal heads respectively;
$p_8, p_9$ - the maximum height of the insole indicated by markers medially and laterally opposite the midheel point.

At this stage the data file was re-orientated by rotation after Foulston et al so that it matched the axes shown in Figure 7.12; the origin is at $p_1$, the y axis is in
Figure 7.12 The marker points added to an orthotic insole data file to establish a model of the form of Figure 7.7, and the subsequent orientation and slicing.

the direction $p_2 - p_5$, the x axis is perpendicular to the y axis such that the vector $p_3 - p_1$ is in the xy plane, and the z axis forms a right-handed orthogonal triple with the x and y axes.

(ii) Selecting Relevant Data Points

Again after Foulston et al, it was decided to take thin slices of the data file in the midfoot region - that is between $p_2$ and $p_5$ - perpendicular to the y axis, where 'thin' meant a slice of thickness 3% of the maximum extent in the y direction of the data, see Figure 7.12. All data points in the slice were projected into the $y = 0$ plane. In the heel region the method was extended to take slices of the same thickness at regular angular intervals about the midheel point $p_5$, such that each slice contained the line parallel to the z axis through $p_5$. The slices were each rotated to be parallel to the midfoot slices and projected into the $y = 0$ plane.
(iii) Fitting a Polynomial Curve to the Data

Having obtained several slices from the data file, an example of which is given in Figure 7.13, with each slice containing points at x,z locations, it would be possible to use the polynomial curve fitting techniques suggested by Bevington or Froberg immediately to give a function of z against x [BEVINGTON 69, FROBERG 81]. However, on inspection of the example, it is apparent that the desired curve could be nearly vertical, at both ends of the range under consideration in the midfoot region, and at one end in the heel region. Polynomial curves do not give good fits in such situations, and are likely to be highly unpredictable. The solution adopted here, and shown in Figure 7.14, was to divide the midfoot slices of data into two overlapping portions so that the curve would be nearly vertical at only one end of the slice and to leave the heel slices as one portion. Each of the portions was rotated by 30° to give a situation where polynomial curve fitting techniques could be employed, and after the fitting of the curve, the portion and the curve were rotated back by the 30°.
(iv) **Reading Positions off the Curve at Required Intervals**
After the curve fitting, vertex positions were read off the curve at regular distances along it such that the first vertex position had the same x value as $p_2$ and $p_3$, and the last vertex position had the same z value as the maximum height of the insole relevant to that slice, determined from $p_6, p_7, p_8, p_9$.

(v) **Projecting the Positions Back into the Original Three Dimensions**
The vertex positions established at regular intervals along the curve fitted to each slice were projected back into the original space by reversing the procedure adopted to project the points into the $y = 0$ plane. The relevant vertices of the mesh were assigned to the positions calculated.

An example mesh calculated by this procedure is given in Figure 7.15. It can be seen from the figure that the distances between vertices adjacent to the vertices of order four in the mid-heel region do not satisfy (6.42). However, (6.42) is a sufficient condition and not a necessary condition for the avoidance of looped surfaces. Here, checking has found that the resulting tangent estimates do not involve strange shapes or a distorted surface, possibly because the surface is
fairly flat in that region. If there were a problem, it would be necessary to move the vertices which cause the problem so that they are closer to the average position of their adjacent vertices. This was not implemented here because the calculation of a row of vertices which are not in a straight line would be much more involved mathematically.

7.5 Remarks

In this chapter, techniques for adapting the new surface model to particular situations and fitting data to the model have been presented. Examples have been shown which arise from situations in prosthetics and orthotics.

The procedure of adapting the new surface model to a particular situation has been broken down into three stages to aid understanding of the considerations involved. Guidelines have been given for determining which portions of the
object being modelled should be represented by which type of rectangular region, and a mechanism given for checking that the developed mesh of vertices is suitable for the new surface model. It has been shown by adaption to the whole foot situation that the mesh for the model may not be unique in any application, and this is because the decision of which regions of the object are best represented by rectangular regions can depend on the conception of the person who designs the mesh. In the whole foot situation, one mesh was developed by Lord and Travis when the foot was considered as ‘sock’ shaped and an alternative was adopted by Gundill and Hackney and Clarks when the foot was considered in two distinguishable sections - ‘sole’ and ‘upper’ [LORD & TRAVIS 89, GUNDILL & HACKNEY 88].

Three procedures for the fitting of data to the model in particular situations were presented. The first, using structured data files, requires no detailed mathematical processing at this stage, but rather relies upon the data capture technique to digitise data which would fit the mesh for the model. One example of this is the UCL CASD system where the digitiser is set up to capture information only at regular angles and regular intervals of pitch. This will not be the general situation, however, and techniques such as dividing up the data and averaging, and fitting data by curve fitting techniques, will often be required. Each of these procedures was again broken down into several stages to aid understanding of the considerations involved. Dividing up the data is easy to visualise, and relatively easy to implement, but as results in the next chapter will show, the final mesh is not necessarily ideal. Using curve techniques is more tricky because of the mathematical processing required in its computer implementation, but, as results will demonstrate, the final mesh is reliable.

For both adapting and fitting, the techniques presented in this chapter are not exhaustive but are given as examples. Similar considerations and techniques can be adopted in other situations.
Chapter 8

Results of Fitting Data to the New Surface Model

8.1 Introduction

8.2 Evaluation of the point-model distance

8.3 The UCL CASD system and the new surface model

8.4 A whole foot model using data averaging techniques

8.5 An orthotic insole model using curve fitting techniques

8.6 Modifications applied to the orthotic insole model

8.7 Remarks
8.1 Introduction

In previous chapters, a new surface model has been developed and techniques presented which enable the model to be adapted to particular situations and data to be fitted to the model. In this chapter, attempts are made to quantify the ability of the model to represent data in prosthetic and orthotic applications by use in three sample situations with different data fitting techniques. Results of how well the data is represented by the surface model are determined by calculation of the shortest distance from each data point to the surface, known as the 'point-model distance'. The technique for calculating the point-model distance is outlined in section 8.2, and in each of the subsequent sections the average, standard deviation and maximum point-model distances are evaluated and a table presented of the spread of the distances.

In section 8.3, results are presented from the use of the new surface model with data from the UCL CASD system. This is for two reasons: first, to determine the suitability of the model for this situation and, secondly, to attempt to quantify the advantages which surface modelling can offer to a system which had not previously involved surface modelling. In section 8.4, the new surface model is applied to a whole foot situation after the data has been fitted by using a dividing up and averaging technique, and in section 8.5, data from an orthotic insole situation is fitted to the model by using a curve fitting method. In section 8.6, modifications to the surface are discussed and a first implementation of the plantar surface eversion quantified by Foulston et al to the orthotic insole of section 8.5 [FOULSTON et al 90] is presented. Finally, conclusions and closing remarks are given in section 8.7.

8.2 Evaluation of the Point-Model Distance

As stated in the previous section, the 'point-model distance' for a data point and a surface model is defined as the distance between the data point and the nearest point on the surface. In this section, the technique for calculating this distance is outlined.
The new surface model is composed of a number of tensor product Bezier patches which can be regarded as independent portions of the surface. For each patch which has at least one of its corners within a threshold distance, perhaps 5cm, of the data point under consideration \( p \), the minimum distance between \( p \) and that patch is calculated. The point-model distance is the least of these distances over all the patches.

The minimum distance between the data point \( p \) and the tensor product Bezier patch \( b(u,v) \) is calculated by using the parameter domain over which the patch is defined, \((u,v)\) where \( 0 \leq u \leq 1, 0 \leq v \leq 1 \). A step size, \( h \), is initially set as 0.5, and \((u,v)\) as (0.5, 0.5). The distance from each of the points \( b(u+h,v+h), b(u+h,v), b(u+h,v-h), b(u,v+h), b(u,v), b(u,v-h), b(u-h,v+h), b(u-h,v), b(u-h,v-h) \) to \( p \) is calculated, and then \((u,v)\) are reset to be the values at the nearest of these points. The procedure is repeated such that if at any stage the nearest point is \( b(u,v) \) then the step size is halved. Only those points with parameter values within the allowed range are considered, and the procedure is repeated until the step size is less than a threshold value, say 0.001. At this
stage the current distance from $b(u,v)$ to $p$ is taken to be the minimum distance from the patch to the data point. A diagram of the first few stages in this iterative procedure is given in Figure 8.1.

This procedure will certainly pick out a local minimum for the point-model distance. If higher order patches were used there would be the possibility of the wrong point and therefore distance being calculated, but with this situation where the order of the Bezier patches is small and the point-model distance is small compared with the overall dimensions of any patch this possibility is extremely remote. None of the tests conducted has indicated such an occurrence.

Each of the tables in this chapter involves the calculation of point-model distances truncated down to the nearest milimetre. This may affect the mean values calculated by up to 0.5 mm, but since all the sets of results are treated in exactly the same manner, consistency is maintained. The standard deviations are not affected significantly by the truncation, and the maximum values are affected by not more than 1 mm.
8 Results of Fitting Data

<table>
<thead>
<tr>
<th>Point-Model Distance (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>r m</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>9 1</td>
</tr>
<tr>
<td>6 1</td>
</tr>
<tr>
<td>4 1</td>
</tr>
<tr>
<td>3 1</td>
</tr>
<tr>
<td>2 1</td>
</tr>
<tr>
<td>6 2</td>
</tr>
<tr>
<td>4 2</td>
</tr>
<tr>
<td>2 2</td>
</tr>
<tr>
<td>9 3</td>
</tr>
<tr>
<td>6 3</td>
</tr>
<tr>
<td>3 3</td>
</tr>
</tbody>
</table>

Table 8.2 The Point-Model Distances for a typical below-knee socket file from the UCL CASD system when fitted to the new surface model.

8.3 The UCL CASD System and the New Surface Model

One of the reasons why the UCL CASD system described in chapter 2 and in [REYNOLDS 88, REYNOLDS & LORD ip] was regarded as a first generation system was that it did not contain an entire knowledge of the surface it designed, but only knew the position of the surface at certain discrete locations. In this section the new surface model is applied to data from the UCL CASD system, thereby giving a model which does know the entire shape of the surface. It was thought that if surface modelling were employed then the total amount of data required by the system might be less than that needed when only discrete positions were known. This possibility is investigated.

The data files for the UCL CASD system contain the positions of points on a cylindrical rectangular mesh at 1/4" intervals between slices and 10" intervals between strips (see Figure 2.11) about a vertical axis. As suggested in chapter 7, once the data is in (x,y,z) format, it is in the correct form for fitting to the new surface model by the techniques of chapter 6 without any further processing.
A set of tests was carried out on a typical below-knee prosthetic socket and the respective measurement file taken from a plaster wrap. Each test involved the fitting of the new surface model and the subsequent calculation of the point-model distances for each point in the file. A below-knee socket and measurement were chosen because the two files are directly comparable, whereas in the above-knee case the measurement file is considerably shorter than the socket because the brim portion is initially a separate file.

First, every $r$th point on every $r$th slice of the measurement and socket were the only points used to define the new surface model, where $r$ takes values 9, 6, 4, 3, 2. The tests involve the calculation of the point-model distance for each of the points in the data file, together with the average, standard deviation and maximum of these distances. Secondly, the procedure is repeated except that $m$ intermediate data points between each $r$th slice and each $r$th strip are used to determine the tangent and twist values for the model, where $m$ takes values 2 and 3. In the tabulated results, a value of $m = 1$ will be used to signify the cases where no intermediate information is used. Tables 8.1 and 8.2 show the point-model distances for the measurement and socket files respectively, and Table 8.3 the average, standard deviation and maximum values. Figures 8.2 and 8.3 show the developed below-knee prosthetic socket model for $(r,m) = (3,3)$ and $(r,m) = (4,1)$ respectively.

The total number of points in the rows of a table can vary because the choice of every $r$th slice may or may not mean that the entire data file is covered. Moreover, the total number of points in the measurement file differs from that in the socket file since a typical socket is extended distally by about 12mm compared with the measurement.

Each of tables 8.1, 8.2 and 8.3 demonstrates the result that the larger the quantity of information used to fit the surface model, the more accurately the model will represent the data file as a whole. This is entirely predictable when the data points are never very close, as in this example, but when there is possible clustering of the points it is likely that the goodness-of-fit will increase up to a certain level as more data is used to generate the model. Generally, the measurement file is more accurately represented by the model than the socket.
Table 8.3 The mean, standard deviation and maximum point-model distances of Tables 8.1 and 8.2.

<table>
<thead>
<tr>
<th></th>
<th>Measurement</th>
<th>Socket</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>r m</td>
<td>Mean (mm)</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2.72</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0.84</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0.25</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.06</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.01</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>0.26</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0.05</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.00</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>1.22</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>0.22</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.01</td>
</tr>
</tbody>
</table>

file is. After determination of the points which are least well represented by the socket model, it is apparent that these points lie in two regions - close to the patella tendon, and in the posterior region of the socket behind the knee. These two areas can be characterised by their significant local variation in shape and curvature. Further investigation may show that modification of the model within the patches using the techniques of section 6.7 may reduce these point-model distances.

Within both the measurement and socket files, there is a further region which is not well represented by the model, and that is the distal end. At this end, the shape varies widely and unpredictably, and this is thought to be due to the difficulty of ensuring smooth operation of the measuring equipment when the radial values it captures are small. The difficulty is caused by friction, but the effects are usually all but eliminated by the manufacture process. Moreover, the
portion of the amputee's residuum in contact with this area of the socket is composed of pliable tissue and, therefore, volume rather than precise shape is of greater importance.

The question still remains, what is a 'good' representation of the data? One suggestion is that each point should have a point-model distance of less than, say, 3 mm. In this case the values of \((r,m)\) which would give 'good' representations of the measurement file are \((2,1), (6,2), (4,2), (2,2), (3,3)\), and of the socket file are \((2,1), (2,2)\) and \((3,3)\). This may be a good criterion when the data file used can be assured not to be ambiguous, and each point is assumed to be accurate to within, say, 1 mm, as may be true in this case. However, if there is a greater variability in the accuracy of the measurement it may be better to use the average and standard deviation figures. In this case, a 'good' representation could be regarded as one where the average and standard deviation of all the point-model distances are both not more than 0.5. Then the values of \((r,m)\) which
give a good representation are: for the measurement (3,1), (2,1), (4,2), (2,2), (6,3),
(3,3), and for the socket (2,1), (2,2), (3,3). If the value of 0.5 is relaxed then more
of the tests yield 'good' representations.

In conclusion, the tables show that it is not necessary to have one Bezier patch
between each set of four adjacent data points. It is suggested that good
representations can be obtained by using one patch for every third slice and
every third strip, but using the intermediate information to calculate the tangent
and twist values, ie \((r,m) = (3,3)\), or by using every fourth slice and strip with
every other slice and strip to determine the tangent and twist information, ie
\((r,m) = (4,2)\). In the latter case, the amount of information required by the surface
model is only a quarter of that required by the UCL CASD system. Therefore,
the surface model is able to introduce significant advantages in data storage
requirements.
8.4 A Whole Foot Model using Data Averaging Techniques

The second application in which the new surface model was used was a whole foot model. This is an extension of the work first presented by Lord & Travis [LORD & TRAVIS 88]. The input data files were composed of \((x, y, z)\) points and captured from a view of the foot by an Isis scanner (see section 3.3) [OXFORD]. Three views were required to scan the whole foot and the three resultant files were aligned graphically and subsequently concatenated. Using the technique for fitting the data outlined in section 7.4.2, the data was fitted to the new surface model. Again, the tests in this section involved calculation and tabulation of the point-model distances between the concatenated data file and the final surface model.

After these results have been tabulated, the point-model distances of the heel and the toe regions are given separately because it was suspected that these regions were particularly poorly represented by the surface model. The positions of the vertices of the mesh were then repositioned graphically using the three-dimensional editor to give a better representation of the original data, and the tests were repeated. Table 8.4 gives the results of all these tests, and Table 8.5 reduces the information to mean, standard deviation and maximum point-model distances. Figure 8.4 shows a computer graphics representation of the whole foot model.

The reason it was suspected that the surface model represented the toe region badly was that the concatenation of the three data files caused confusion in that area; that is, using the three-dimensional editor, it was difficult to determine the shape of the surface in that region. This was partly because the errors in the positions of the data points were particularly significant in that region, and partly because there were not enough data points within each divided-up portion of the file to accurately estimate the vertex position. As can be seen from the tables, the representation of the toe data was improved when the vertex points were positioned graphically.
Table 8.4 The Point-Model Distances for a whole foot model fitted to the new surface model by data averaging (a) without modifications and (b) with the heel and toe region vertices modified graphically. The results from the heel and toe regions are included in the table.

<table>
<thead>
<tr>
<th></th>
<th>Without modifications</th>
<th>With heel and toe modifications</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean (mm)</td>
<td>Std Dev (mm)</td>
</tr>
<tr>
<td></td>
<td>Mean (mm)</td>
<td>Std Dev (mm)</td>
</tr>
<tr>
<td>Foot</td>
<td>0.98</td>
<td>1.60</td>
</tr>
<tr>
<td>Toe</td>
<td>1.91</td>
<td>2.50</td>
</tr>
<tr>
<td>Heel</td>
<td>2.30</td>
<td>1.89</td>
</tr>
</tbody>
</table>

Table 8.5 The mean, standard deviation and maximum point-model distances of Table 8.4.

The reason that poor representation was suspected in the heel region was that the particular mesh of surface model chosen leads to very few vertex positions in that region. This meant that the positions calculated were not very representative of the region because the shape varied too much between the vertex points. Again, graphical alteration of the vertex points in this region improved the representation of the data in the region.
The model chosen in this section is not a good representation of the data and this can be clearly seen by the size of the mean point-model distance in the heel and toe regions - in both cases it is over 1.50 mm. This indicates two deficiencies in the application: first, the dividing up and averaging technique requires the data points to be a dense cover of the entire surface such that in every 2 mm², say, of the surface there is a data point; second, this model is not ideal for the foot because in the heel region the vertex points are too scarce. A better mesh for the foot might be the alternative presented in section 7.2.2 which is possibly better suited to last design because the boundary between its two rectangular regions can be associated with the feather edge between sole and upper of a shoe. The model would then agree with the technique used by Clarks to design standard shoe lasts and associated patterns.
8.5 An Orthotic Insole Model using Curve Fitting Techniques

The third application to which the new surface model was applied was an orthotic insole model. The data was acquired by an Isotrak digitiser [ISOTRAK] in free streaming mode, as described in section 3.3.2, and the curve fitting technique described in section 7.4.3 was employed to yield the vertex points of the surface model.

In Figure 7.7 the mesh size parameters \( m_{\text{Side}} \), \( m_{\text{End}} \), \( m_{\text{Edge}} \) which describe exactly the size of the orthotic insole mesh, were diagramatically explained. \( m_{\text{Side}} \) is the number of slices between the second metatarsal head and the midheel point, \( m_{\text{End}} \) is the number rows of points medially-laterally across the sole of the foot and \( m_{\text{Edge}} \) is the number of rows of points on the mesh up the side of the foot. The point-model distances are tabulated in Table 8.6 for various values of these parameters, and the resulting average, standard deviation and maximum values are given in Table 8.7. Figure 8.6 shows an orthotic insole over a mesh with parameters \((m_{\text{Side}}, m_{\text{End}}, m_{\text{Edge}}) = (8, 4, 2)\).
Table 8.6 The Point-Model distances for an orthotic insole file fitted by curve fitting techniques.

<table>
<thead>
<tr>
<th>mSide</th>
<th>mEnd</th>
<th>mEdge</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8+</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>4</td>
<td>3</td>
<td>545</td>
<td>126</td>
<td>32</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>2</td>
<td>537</td>
<td>133</td>
<td>30</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>1</td>
<td>497</td>
<td>152</td>
<td>43</td>
<td>11</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>2</td>
<td>513</td>
<td>145</td>
<td>37</td>
<td>11</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

The tables show that for all the values of the size parameters, the numbers of points with higher point-model distances are similar. Examination of the data file reveals that in each case the points are the same, and they are all erroneous data points which have been digitised in positions which are inconsistent with their surrounding data points. They are, therefore, exactly the points which it is hoped that the surface model would not interpolate. It appears that the curve fitting technique has ensured that the inconsistent points are ignored since the vast majority of the points are consistent. Nevertheless, the average and standard deviation of the point model distances are low. If the erroneous points are ignored, there is a further significant improvement upon these values. The low values seem to indicate that the surface model is a 'good' fit to the data.

The fact that the maximum point-model distance is constant for all the values of the parameters, and that this is due to erroneous data points demonstrates that this maximum value is not a good indication of whether the surface model is a 'good' fit. The way which has most clearly demonstrated that the model in this case is a 'good' fit to the data is by turning the data points with large point-model distances into marker points and re-examining the data file with the aid of the three-dimensional data editor of chapter 4. An example of a slice containing one of the points with point-model distance 7 mm is shown in Figure 8.5.
Table 8.7 The mean, standard deviation and average Point-Model distances for the orthotic insole model tabulated in Table 8.6.

8.6 Modifications Applied to the Orthotic Insole Model

Chapter 6 developed a new surface model together with suggested techniques for modification of the model in section 6.7. However, in order to progress from the mathematical ability to perform modifications on the surface, to the ability to manipulate the surface in a clinically desirable manner requires an understanding of how, biomechanically, the features of the body segment move and behave relative to one another.

D'Arcy Thompson examined the relationships between shapes of the anatomies and forms of a variety of living creatures [THOMPSON 68]. Waters has developed a model of how the surface of the face reacts to the tensing of an underlying muscle, and Foulston et al have quantified the eversion of the plantar surface of the foot [WATERS 87, FOULSTON et al 90]. Reynolds has investigated the stiffness of tissue in the lower leg residuum [REYNOLDS 88, REYNOLDS & LORD ip], and the UCL system has required research into the change of shape between an amputee’s residuum and a desirable socket shape. These are indicative of the kinds of study which may be necessary to develop clinically desirable modifications to the surface model for particular applications.

In this section, the changes in shape caused by plantar surface eversion quantified by Foulston et al are applied to the model of the previous section. This is at present only a preliminary application and is presented to indicate the
features available to surface modelling. In particular the surface slope changes in [FOULSTON et al 90] would require further investigation to establish the point about which the plantar surface rotates before a definitive application is achieved. Here it has been assumed that the rotation is about the x axis which is approximately along the first metatarsal (see section 7.4.3 and Figure 7.12). Figure 8.7 shows the orthotic insole with the eversion affected, and can be compared with Figure 8.6. Further investigation to enhance the result might include, for example, evaluation of the point-model distances between the model based on a data file with no eversion, but where the eversion is applied to the model, and a data file captured where eversion is already apparent.

8.7 Remarks

This chapter has presented results of tests fitting the new surface model to data sets in three different prosthetic and orthotic applications with three different data fitting techniques.
In the first situation, the model has been fitted to data from the UCL CASD system using different number of patches and different quantities of the available data. The new surface model is capable of giving a ‘good’ representation of the shape, and it was hypothesised that, if the modification techniques were related to clinical procedure, the surface model for the prosthetic socket would be adequate with only a quarter of the information currently used by the UCL CASD system and other typical prosthetic socket design systems.

The second situation involved the fitting of data to a whole foot model using a dividing up and averaging technique. The original data files proved to be unsatisfactory, and this may indicate that visual scanners may require allowances to be made for their accuracy. The dividing up and averaging technique also gave unreliable results. Furthermore, the ‘sock’ shape mesh adopted for the whole foot did not give a good representation in the toe and
heel regions, and therefore, for shoe design applications it is probably an advantage to use a 'sole and upper' mesh which would be an extension of that developed by Clarks for shoe pattern design.

The third situation was an orthotic insole model, where curve fitting techniques proved very satisfactory in fitting data to the model. The technique ignored occasional erroneous data points. This technique, it is suggested, should be transferrable to the whole foot situation for shoe design once a suitable pattern for curve fitting to the whole foot has been devised.

In summary, the results of this chapter indicate that the new surface model can give a good representation of a body segment for prosthetic and orthotic application provided that a reliable method for fitting the data to the model is available. Two such methods are, first, capturing only the required information, and, second, curve fitting techniques.
Chapter 9

MODEL - The Surface Display Software

9.1 Introduction
9.2 Format of the data file
9.3 Surface display modes
   9.3.1 Wire frame display
   9.3.2 Shaded solid views
9.4 Remarks
9.1 Introduction

In the use of computer aided design systems, the ability to view a designed object and examine it before manufacture has always been a necessary and attractive feature. Examination means that many mistakes, inaccuracies and unaesthetic shapes can be spotted and corrected before the object is committed to the manufacture process. In the context of generally accepted principles of surface display, this chapter describes the program MODEL, which is the software developed as part of this project for viewing surfaces comprised of the new surface model. It will in fact display a general surface comprised of quartic Bezier patches, and will generate the display in two distinct modes: wire frame and shaded solid view. Calculation of the surface uses the methods presented in chapter 5. Section 9.2 documents the format of the required data file, and section 9.3 discusses the display modes, giving examples of the prosthetic socket, whole foot and orthotic insole models developed in the previous chapters. Further remarks in section 9.4 conclude the chapter.

9.2 Format of the Data File

The data file, principally for reasons of readability, is an ASCII file containing a series of quartic Bezier patches, the format of which is given in Table 9.1. The first line for each patch gives the number of that patch, and the second contains the numbers of the vertices at the corners of the patch in clockwise order; that is, clockwise when the patch is viewed from outside of the surface. These first two lines for the patch are not strictly required, but are useful when ensuring that modifications are applied to the correct portions of the surface model. Thereafter the Bezier points of the patch are listed in the order suggested. There is no header for the file, and termination is by an 'end of file' character. Often many of the Bezier patches will only be cubic in degree, and so the storage required for the patch when raised in degree to a quartic patch is greater than strictly necessary. However, when the patches are stored as quartic patches, the surface manipulation techniques given in chapter 6 are available without further storage requirements.
where $P_k$, an Integer, is the number of patch $k$, $C_1, C_2, C_3, C_4$, all Integers, are the numbers of the points which are the corners to the patch, $x_{ij}, y_{ij}, z_{ij}$ are the $x,y,z$ coordinates of Bezier point $ij$ of the patch, each being a Real number, and $<WS>, <EOL>$ are respectively ‘white space’ (usually a Tab or Space) and ‘end of line’ characters.

Table 9.1 The format of one Bezier patch contained in a data file for program MODEL.
9.3 Surface Display Modes

The two surface display modes which the program MODEL offers, described here, are the generally accepted and used modes in engineering, see for example, [FOLEY & VANDAM 82, FORREST 79, NEWMAN & SPROULL 81]. Wire frame display is usually used to give a first impression while shaded solid views provide a more precise understanding of the shape.

9.3.1 Wire Frame Display

The first, common, method for displaying surfaces is by way of a wire frame display. This mode draws up the boundaries to each of the Bezier patches and this is, in practice, the same as the mesh connecting each of the vertices of the mesh used when constructing the surface model. The program MODEL displays wire mesh surfaces in two different styles, first with all lines shown and second with hidden lines removed. An example of the style of all lines being shown is given in Figure 9.1, which is taken from the prosthetic socket model of the

Figure 9.1 A prosthetic socket model displayed in wire frame mode with all lines visible.
previous chapter. The view can be confusing with so many lines shown since their relationship to one another in three-dimensional space is not immediately clear. Rotating the object can help to reduce any misconceptions, but an alternative style is to show the wire frame with the lines which would be hidden in a solid view removed. The same prosthetic socket is given in this second style in Figure 9.2. It is easier, in this style, to visualise the shape of the final surface with less danger of misunderstandings, and the time taken to draw the second style is not significantly greater than the first in most applications - wire frame displays are quick and easy to draw. The hidden line removed style is usually the preferred style of a wire frame display although it does not actually portray as much information as the complete wire frame style.
Figure 9.3 The normal $n$ to a sub-patch is the cross product of its two diagonals $r_2 - r_1$ and $r_4 - r_3$. The intensity with which the sub-patch is shaded is determined by the cosine of the angle between its normal and the vector $I$ to the light source.

9.3.2 Shaded Solid Views

It is generally accepted that wire frame views can be confusing, and often do not portray easily all the shape information of a surface since they only display the relative positions of certain discrete points. An alternative is to use a shaded solid view of the surface. A thorough investigation of shading techniques can be found in Foley and van Dam or Newman and Sproull [FOLEY & VANDAM 82, NEWMAN & SPROULL 81], but the technique adopted by the program MODEL is given briefly here.

When a shaded solid view is chosen during execution of the program, the user chooses a density value between 1 and 15. For each patch the program calculates the position of points on the surface of the patch at density $\times$ density regular intervals of the $(u,v)$ parameter domain, giving density $\times$ density sub-patches. A normal vector is then worked out for each of the sub-patches by supposing that
the sub-patch has corners \( r_1, r_2, r_3, r_4 \) in clockwise cyclic order when viewed from outside of the surface, and estimating the unit normal vector as \( n \), see Figure 9.3, where

\[
n = \text{unit}((r_2 - r_4) \times (r_1 - r_3)) \tag{9.1}
\]

Intensity value 0.0 is assigned to black, 1.0 to white, and an ambient light intensity of \( a \) is assumed, where \( 0.0 < a < 1.0 \). A light source is assumed to be at a large distance in direction \( l \) where \( l \) is a unit vector, and then the intensity value for the sub-patch is given by

\[
I = a + (1.0 - a) \times \max\{0, (l \cdot n)\} \tag{9.2}
\]

Because both of the vectors in the scalar product of this equation are unit vectors, the scalar product is identically equal to the cosine of the angle between the two vectors, Figure 9.3. Therefore, (9.2) gives a value \( a \leq I \leq 1.0 \), and the whole of the sub-patch is filled with a grey colour of this intensity. All the figures given in this chapter assume a value \( a = 0.3 \), a value determined by experiment. To increase the reality of the image, and lessen the effect of the edges of the sub-patches being discernible, the value of density can be increased. The edges are noticeable because the eye picks up the jump in intensity at the joins of the sub-patches. Newman and Sproull and Foley and van Dam suggest two shading techniques, namely Gouraud and Phong shading, where the shade across each sub-patch is varied to enhance the realism of the image. Although the techniques do not affect the smoothness or shape of the object in any way, they make the object appear smoother and more pleasing. Forrest suggests the use of 'non-realistic' shading to highlight the curvature properties of the surface. This may be applicable if curvature proves to be of critical importance, but Gouraud, Phong and 'non-realistic' shading techniques were not applied in the program MODEL because the time taken in the extra calculation on the available computer hardware was adjudged, after experimentation, to be unacceptable.

A common method for hardware to display coloured images is to use a look-up table of 256 colours. The colour to be used at any point in the presented image is indicated by the choice of a number in the range 0-255, and the red, green and
blue intensity values of that particular colour are contained in the table. The program MODEL sets aside a number of consecutive colours which are given grey values ranging from black to white, and these are the colours used to shade the surface. It was found that 64 colours was a sufficient range for the differences between successive colours to be indistinguishable.

Examples of solid shaded views are given in Figures 9.4, 9.5, 8.4 and 8.6. Figures 9.4 and 9.5 show the prosthetic socket model with different values of density, while Figures 8.4 and 8.6 show shaded solid views of the whole foot and orthotic insole models respectively.
9.4 Remarks

The software program presented here, MODEL, was developed to give graphical images of the shapes developed in this project. It is a program which is able to show Bezier patch surfaces in either of the standard wire frame or shaded solid modes. The program also includes the ability to rotate the object so that the surface seen from any angle can be displayed, thereby enhancing the understanding of the shape of the surface, as in the data editor of Chapter 4. In wire frame mode, the display can include or exclude hidden lines and the rotation can be in real time, but the hardware available was not sufficient to allow real time rotation of shaded solid images. Shaded solid models give a good visual indication of the shape of the surface before the object is committed to manufacture.
Chapter 10

Conclusions

10.1 Conclusions from the current work

10.2 Suggestions for future work
10.1 Conclusions from the Current Work

At the outset of this thesis, the UCL Computer Aided Socket Design system for the design of prosthetic sockets was described, and reasons why the system was regarded as a First Generation system were given. The intention of developing a Second Generation system for prosthetics and orthotics, which did not have the restrictions of limited applicability and knowledge of the designed shape only at discrete locations, was stated. After the documentation of the UCL CASD system for above-knee amputees in Chapter 2, the rest of the thesis has been dedicated to this intention. The result is not a Second Generation system in its final form, but rather a new surface model and indications of a number of other features which such a system should include. Figure 1.1, which showed the constituent parts of an integrated CAD/CAM system for prosthetics and orthotics, is reproduced as Figure 10.1, but here the relevance of the work in the various chapters of this thesis is indicated in the figure.

In Chapter 3, the form of the data to be expected from a data capturing technique was explored. While, occasionally, especially with a hand held tactile sensor, it may be possible to capture only the information required by the employed surface modelling technique, a general CAD/CAM system for prosthetics and orthotics should be able to handle a file of (x,y,z) data points covering the entire body segment to a known density. The introduction of surface modelling to the UCL CASD system in sections 7.4.1 and 8.3 demonstrated the advantage in ease of application of the surface model, when only the minimum necessary information for the surface is captured. However, the difficulties of specifying exactly what information constitutes the minimum required and how to capture it meant that this feature could not be assumed in every application.

An editor for graphical examination and alteration of data files was presented in Chapter 4. A particularly useful feature was the ability to distinguish between two types of points: data points and marker points. Subsequently, marker points were found necessary when fitting the captured data to the surface mode in chapter 7, and were also helpful when analysing how well the final model represented the captured data in chapter 8.
Chapter 5 gave a review of surface modelling techniques currently used in engineering and other fields, with particular emphasis on how the techniques are used to interpolate a number of known points in three dimensions. The aim was to find a technique which could be used in many applications arising in prosthetics and orthotics, yet which contained sufficient flexibility to allow the modifications to component shape which are clinically required during their design. None of the techniques was considered suitable, mainly because most of them require a 'rectangular' situation for implementation, and such situations
are rare in prosthetics and orthotics. However, it was concluded that
development of a new surface model should use tensor product Bezier patch
techniques in 'largely rectangular' situations.

Following on from the review and conclusions of chapter 5, a new surface model
for general application in prosthetics and orthotics was developed in Chapter
6. Precise requirements for its application together with explicit formulae for its
implementation and techniques for modifications of its shape were established.

The implementation of the new surface model developed in chapter 6, with data
files of the form suggested by the review of data capturing methods in chapter
3, was examined in Chapter 7. The procedure for adapting the new surface model
to a particular situation was broken down into several steps, and implemented
for the prosthetic socket, the whole foot and the orthotic insole. Various methods
for fitting the data file to the new surface model were demonstrated and, again,
broken down into several steps. The easiest implementation was when the data
captured was exactly that required by the adapted new surface model. However,
as discussed earlier, this situation cannot be expected in general, and so other
methods, one involving the dividing up of the data and averaging, and the other
employing curve fitting techniques, were also investigated. Each of these
methods was also broken down into several steps and implemented in a specific
situation. While the methods were not exhaustive, they indicated considerations
and procedures which would be relevant in other applications.

Experimental results were given in Chapter 8 of how well the new surface model
represented the original data in three applications. Each application involved a
different method for fitting the data to the model. In general, up to a limit, the
more Bezier patches the new surface model was composed of, the more
accurately the original data was represented; the price for more Bezier patches
was that the model took a longer time for calculation and display, and had a
greater data storage requirement. For a prosthetic socket, using data from the
UCL CASD program, it was suggested that adequate representation might be
achieved by using only a quarter of the information currently stored; that is
positions on slices at half inch spacing and strips at 20° intervals. From
application to the whole foot, it was apparent that considering the foot as a 'sock'
shape was not satisfactory since it lead to poor representation in the heel region. The preferred shape is ‘sole and upper’, which is a direct extension of the model used by Clarks, for example, in the design of shoe patterns for standard lasts. In the whole foot application, poor representation was also evident in the toe region. This was because the dividing up and averaging technique for fitting data to the new surface model was unsatisfactory, both when there were erroneous points in the data file, and when data was scarce in some region. When either of these situations occurs, an alternative method of data fitting should be used. Application of the new surface model to the orthotic insole employed curve fitting techniques to fit the data. This overcame the shortcomings of dividing up and averaging, and had the effect of ignoring erroneous data points when they were inconsistent with the surrounding points. Results of modelling the insole were very satisfactory, and suggest that, for future shoe applications, a curve fitting technique should be devised and employed with the ‘sole and upper’ surface model.

Chapter 8 also briefly looked at the suitability of the new surface model for the modifications of component shape which are required for clinical reasons, and, why it is important for a CAD system for prosthetics and orthotics to include this feature. A first implementation was applied to eversion of the orthotic insole model, after the biomechanical movements were reduced to information suitable for a CAD system.

Finally, because surface display and visualisation before manufacture can indicate problems and eliminate wasted time, effort and materials, the surface display tool developed as part of this project was discussed in Chapter 9. It was concluded that a CAD system should include at least two modes of surface display: wire frame and solid shaded views. There should also be the ability to view the component from any angle.

In summary, this project has developed a new surface model for general application in prosthetics and orthotics, with tools for its implementation, and has investigated a number of other features which it is suggested that a Second Generation CAD system for prosthetics and orthotics should include.
10.2 Suggestions for Future Work

In the shorter term there are several avenues for further work to which this thesis leads:-

The surface modelling technique developed here has been shown to give satisfactory results by using graphical computer visualisations and other computer analysis of the surface shape compared with the original data. The testing by computer could be extended to include other techniques used in the engineering industries, such as 'reflection lines'; 'reflection lines' is a technique in which the simulation of a series of strip lights is reflected off the surface to give a reflected pattern. It can reveal subtle variations in shape, but requires a greater amount of computer power than was available to this project. However, since the final acceptance of any design process in prosthetics and orthotics depends upon the reliability of a manufactured item, the modelling techniques developed here should be extended, using existing methods for the manufacture of Bezier surfaces, to give physical components. Physical objects can be investigated by touch, and it may be that smoothness features are more apparent by touch than sight. It is therefore possible that some applications may desire curvature continuity in addition to the tangent continuity of the model presented here. This would require further development of the new surface model.

It was stated, in the analysis of the application of the new surface model to data from the UCL CASD system, that an understanding of the relationship between rectification patterns, brim shapes and the modifications available to the new surface model may improve those results. The understanding could be developed in the same manner as the initial rectification patterns were introduced to the system; that is, a number of patients could be treated by a prosthетist in a conventional manner, and the resulting rectifications quantified in terms of the model.

In a footwear application, a 'sole and upper' philosophy has been suggested for the surface model. However, further work is required to establish a curve fitting technique to fit the captured data to the new surface model. The subsequent
stage in the process of shoe design, that is modification of the foot shape to a
last shape, also needs further investigation before a complete understanding is
achieved.

Preliminary modifications have been applied to an orthotic insole model as the
result of a biomechanical study. However, even here, further investigation is
required to determine how the plantar surface eversion affects the shape of the
insole. The understanding could be tested, for example, by calculating the
point-model distances discussed in chapter 8 for a data file of an everted plantar
surface, compared with a surface model of a plantar surface with no eversion
to which the eversion had subsequently been applied.

In the longer term, there is much discussion about an integrated,
all-encompassing, CAD/CAM system for prosthetics and orthotics, possibly
built directly on the experience of systems such as UCL CASD. The system will
have several modules common to all applications, such as the measurement and
manufacture facilities, and a further expert system for each individual
application. The previous three paragraphs have just indicated the first steps in
two specific applications towards the establishment of expert systems which
will aid the operator of the system. The major features highlighted in this thesis,
such as the form of the data file, a data file editor, surface modelling, data fitting
techniques and surface display software, should be considered in the philosophy
and design of such an integrated system.
List of the Author’s Publications


References


[IMSL] *MATH/LIBRARY* Software Package, IMSL. Tel: (USA) (713) 782 6060.

[ISOTRAK] *3Space Isotrak*, Polhemus Navigation Sciences Division, McDonnel-Douglas Electronics Company, Vermont USA. Tel: (USA) (802) 655 3159.


[LEACH 88] M.O.Leach (1988), Spatially localised nuclear magnetic resonance, In [WEBB 88a].


Computer-aided design and pattern generation for the
garment industry, Review no 2, Clothing Technology
Centre.

[MOFFAT & MIKHAIL 80] F.H.Moffat, E.M.Mikhail (1980), Photogrammetry,
3rd edn, Harper and Row.

Scale-based description and recognition of planar
curves and two-dimensional shapes, IEEE Transaction
on Pattern Analysis and Machine Intelligence, vol 8, no 1.

D.James (1988), A computer system for the interactive
planning and prediction of maxillofacial surgery,
American J Orthodontics and Dentofacial Orthopaedics, vol
94, no 6.


[OXFORD] Isis, Oxford Metrics, Oxford, UK. Tel: (0865) 244656.

[PERCEPT] Perceptor. Micro Control Systems, Vernon, Connecticut,
USA.

[PIEGL 89a] L.Piegl (1989), Modifying the shape of rational
no 8.

[PIEGL 89b] L.Piegl (1989), Modifying the shape of rational
21 no 9.


[PUGH 86b] A.Pugh (1986), Introduction to [PUGH 86a].


[RIOUX 86] M. Rioux (1986), Laser range finder based on synchronised scanners, In [PUGH 86a].


[SHAPE] System Shape developed by Shape Products Ltd., Horsted Keynes Industrial Park, Haywards Heath, West Sussex RH17 7BA. Tel: (0342) 810444.

[SI]  *B-Spline Modelling Package* developed by Centre for Industrial Research, Oslo, Norway. Tel: (Norway) (02) 45 20 10.


[VISION 3D] 3D VideoLaser developed by 3DVision. Tel: (France) (16) 61 54 33 47.


