Copyright Statement

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

This thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author's right to be identified as the author of this thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from their thesis.

To request permissions please use the Feedback form on our webpage. http://researchspace.auckland.ac.nz/feedback

General copyright and disclaimer

In addition to the above conditions, authors give their consent for the digital copy of their work to be used subject to the conditions specified on the Library Thesis Consent Form.
A Toolkit for the Visualization of Tensor Fields in Biomedical Finite Element Models

Burkhard Claus Wünsche

A dissertation submitted to the Department of Computer Science of the University of Auckland in partial fulfillment of the requirements for the degree of doctor of philosophy

University of Auckland
May 13, 2004
This thesis was assessed and approved by

Dr. Richard Lobb
Director of the Graphics Group, Department of Computer Science,
University of Auckland, Auckland, New Zealand

Dr. David Kenwright
Senior Research Scientist, Industrial Research Limited, Auckland, New Zealand

Prof. Dr. Christopher R. Johnson
Director, School of Computing
Director, Scientific Computing and Imaging Institute (SCI)
Director, Center for Bioelectric Field Modeling, Simulation, and Visualization (NIH NCRR)
Director, Engineering Scholars Program (ESP)
Co-Director, Advanced Visualization Technology Center (AVTC)
Distinguished Professor of Computer Science
Research Professor of Bioengineering
Adjunct Professor of Physics
Faculty Member, Computational Engineering and Science (CES) Program
Co-Founder, Visual Influence Inc.

Copyright © 2004 by Burkhard Wünsche
Abstract

Medical imaging is an essential tool for improving the diagnoses, understanding and treatment of a large variety of diseases. Over the last century technology has advanced from the discovery of x-rays to a variety of 3D imaging tools such as magnetic resonance imaging, computed tomography, positron emission tomography and ultrasonography.

As a consequence the size and complexity of medical data sets has increased tremendously making it ever more difficult to understand, analyze, compare and communicate this data. Visualization is an attempt to simplify these tasks according to the motto “An image says more than a thousand words”.

This thesis introduces a toolkit for visualizing biomedical data sets with a particular emphasis on second-order tensors, which are mathematically described by matrices and can be used to express complex tissue properties such as material deformation and water diffusion. The toolkit has a modular design which facilitates the comparison and exploration of multiple data sets. A novel field data structure allows the interactive creation of new measures and boolean filters are introduced as a universal visualization tool. Various new visualization methods are presented including new colour mapping techniques, ellipsoid-based textures and a line integral convolution texture for visualizing tensor fields.

To motivate the design and to assist in the use of the toolkit, guidelines for creating effective visualizations are derived by using perceptual concepts from cognitive science. A new classification for visual attributes according to representational accuracy, perceptual dimension and spatial requirements is presented and the results are used to derive values for the information content and information density of each attribute. A review and a classification of visualization icons completes the theoretical background.

The thesis concludes with two case studies. In the first case study the toolkit is used to visualize the strain tensor field in a healthy and a diseased human left ventricle. New insight into the cardiac mechanics is obtained by applying and modifying techniques traditionally used in solid mechanics and computational fluid dynamics.

The second case study explores ways to obtain in vivo information of the brain anatomy by visualizing and systematically exploring Diffusion Tensor Imaging (DTI) data. Three new techniques for the visualization of DTI data are presented: Barycentric colour maps allow an integrated view of different types of diffusion anisotropy.
Ellipsoid-based textures and Anisotropy Modulated Line Integral Convolution create images segmented by tissue type and incorporating a texture representing the 3D orientation of nerve fibers. The effectiveness of the exploration approach and the new visualization techniques are demonstrated by identifying various anatomical structures and features from a diffusion tensor data set of a healthy brain.
Acknowledgment

A large number of people have helped in various ways during the work for this thesis. I would like to thank in particular the following people:

- My supervisor, Dr. Richard Lobb, who provided essential advice, guidance, and encouragement.

- Prof. Peter J. Hunter and Assoc. Prof. Andrew J. Pullan from the Bioengineering Group of the University of Auckland, Auckland, New Zealand, for valuable discussions and for providing us with a numerical model of a dog heart.

- Dr. Alistair A. Young from the Department of Physiology of the University of Auckland, Auckland, New Zealand, for valuable discussions and for providing us with the models of the human left ventricle.

- Dr. Peter J. Basser from the National Institute of Health, Bethesda, Maryland, USA, for valuable discussions and Dr. Carlo Pierpaoli for providing us with the diffusion tensor data set of a healthy brain.

- Dr. Richard White of the Cleveland Clinic, Cleveland, Ohio, USA, who kindly provided tagged MRI data of a human left ventricle diagnosed with dilated cardiomyopathy.

- Prof. Richard L. M. Faull from the Department of Anatomy with Radiology of the University of Auckland, Auckland, New Zealand, for discussing with us the neuroanatomy of the brain and the resulting visualizations of it.

- Prof. Dr. Hans Hagen and Dr. Dieter Laaser from the University of Kaiserslautern, Kaiserslautern, Germany, for introducing me to the fields of Computer Graphics and CAGD.

- The staff in the Computer Science Department for providing advice and needed resources and making the workplace environment most pleasant.

- The University of Auckland for employing me since January 2001 as a full time lecturer and for allowing me to complete my PhD in the (rare) teaching free periods.
• And last but not least my family and friends for their support during my years at the University of Auckland.
# Contents

Abstract i

Acknowledgements iii

Contents v

List of Figures xi

List of Tables xxi

1 Introduction 1
   1.1 Motivation ................................ 1
   1.2 Contributions ................................ 2
   1.3 Thesis Overview ............................. 3

2 Strain, Stress, and Finite Element Modelling 5
   2.1 Notations and Definitions ....................... 6
       2.1.1 Tensors ................................ 6
   2.2 Curvilinear Coordinates ....................... 8
   2.3 Linear Elasticity ............................ 9
       2.3.1 Displacement and Strain .................... 9
       2.3.2 Stress ................................ 12
       2.3.3 Model of a Linear Elastic Solid ............ 14
   2.4 The Finite Element Method ...................... 16
       2.4.1 Concepts ............................. 17
       2.4.2 Finite Element Approximation ............... 18
       2.4.3 Isoparametric Mapping ................... 25
       2.4.4 Gaussian Quadrature ..................... 27
       2.4.5 Solving a FE Problem ..................... 28

3 Biomedical Structures, Models and Data Sets 31
   3.1 Introduction ................................ 31
   3.2 The Human Heart .............................. 32
Contents

3.2.1 The Structure and the Functioning of the Heart ........ 32
3.2.2 Heart Diseases and Heart Failure .................. 38
3.2.3 Cardiac Imaging .................................. 39
3.2.4 Myocardial Strain and Stress as an Indicator of Heart Failure 42
3.2.5 Measurement of Myocardial Strain and Stress .......... 44
3.2.6 Previous Studies of Myocardial Strain ............... 49
3.2.7 The Visualization of Myocardial Strain and other Functional Values ....................................... 51
3.2.8 FE Modelling of the Heart .......................... 53
3.3 The Human Brain ...................................... 57
3.3.1 The Anatomy and Physiology of the Brain .......... 58
3.3.2 Diffusion Tensor Imaging ............................ 62
3.3.3 Derived Quantities .................................. 64
3.3.4 Analysis and Visualization of DTI Data ............ 65
3.3.5 FE Model for a DTI Data Set of the Human Brain .... 68

4 Scientific Visualization ..................................... 69
4.1 Challenges in the Visualization of Multidimensional Data ........ 70
4.2 A Visualization Schema .................................. 71
4.3 Data Transformation ...................................... 73
  4.3.1 Transformations Based on the Independent Variables .... 73
  4.3.2 Transformations Based on all Variables ................ 73
  4.3.3 Transformations based on the Dependent Variable .... 75
4.4 Human Visual Perception .................................. 76
  4.4.1 Anatomy and Physiology of the Eye ................. 76
  4.4.2 Visual Processing ................................... 77
  4.4.3 Visual Attributes ................................... 78
  4.4.4 Colour Perception and Colour Spaces ............... 80
  4.4.5 Classification of Visual Attributes ................ 82
4.5 Visualization Icons .................................... 83
4.6 Scalar Icons ........................................... 84
  4.6.1 Colour Mapping ...................................... 84
  4.6.2 Contours and Isosurfaces ........................... 88
  4.6.3 Textures ........................................... 89
  4.6.4 Particle Systems ................................... 89
  4.6.5 Height Fields ...................................... 90
  4.6.6 Direct Volume Rendering ............................ 90
  4.6.7 Scalar Field Topology and Features ................ 92
  4.6.8 Visualization Techniques for Multivariate Data .... 93
4.7 Vector Icons ........................................... 94
  4.7.1 Vector Glyphs ....................................... 95
  4.7.2 Particle Advection ................................... 97
  4.7.3 Integral Curves and Surfaces ....................... 98
  4.7.4 Multivariate Colour Maps ........................... 102
  4.7.5 Texture-based Methods .............................. 103
4.7.6  Line Integral Convolution .................................. 105  
4.7.7  Direct Volume Rendering .................................. 109  
4.7.8  Vector Field Topology and Features ....................... 109  
4.8  Tensor Icons .................................................. 113  
4.8.1  Tensor Glyphs .............................................. 113  
4.8.2  Integral Representations .................................. 115  
4.8.3  Surface and Volume Representations ...................... 116  
4.8.4  Tensor Field Topology .................................... 117  
4.9  The Classification of Visualization Icons .................... 118  
4.10 Using Visualization Icons .................................... 122  
4.10.1 Combining Visualization Icons ............................. 125  
4.11 Increasing the Effectiveness of a Visualization ................ 127  
4.11.1 Lighting .................................................. 127  
4.11.2 Perceptual Clues .......................................... 128  
4.11.3 Exploration Techniques ................................. 128  
5  The Visualization Toolkit .......................................... 131  
5.1  Visualization Environments .................................... 132  
5.2  Top Level Design of the Visualization Toolkit ................ 134  
5.3  A Field Data Structure for Interactively Exploring Biomedical Data  
  Sets ....................................................... 135  
  5.3.1 Field Data Structure .................................... 137  
  5.3.2 Graphical User Interface ................................ 140  
5.4  User Interaction in 3D .......................................... 141  
  5.4.1 View & Model Transformations ............................ 141  
  5.4.2 3D Object Selection and Interaction ..................... 143  
  5.4.3 The Field Probe ......................................... 146  
5.5  The Colour Map Control ....................................... 146  
  5.5.1 Colour Spectra ........................................... 148  
  5.5.2 Texture Mapping .......................................... 150  
  5.5.3 Features ................................................ 151  
  5.5.4 Implementation ......................................... 153  
5.6  Volume, Surface and Point Selection .......................... 154  
  5.6.1 Volume Selection ....................................... 155  
  5.6.2 Surface Selection ....................................... 156  
  5.6.3 Point Selection ......................................... 159  
5.7  Filters ........................................................ 163  
5.8  Model Geometry ................................................ 164  
  5.8.1 Visualizing the FE Model Geometry ........................ 164  
  5.8.2 Computing Model Properties from the FE Geometry ...... 166  
5.9  Visualization Icons ............................................ 170  
  5.9.1 Scalar Icons ............................................. 170  
  5.9.2 Vector Icons ............................................. 179  
  5.9.3 Tensor Icons ............................................ 186  
5.10 Additional Features ............................................. 190
## Contents

5.10.1 Contextual Cues .................................. 190  
5.10.2 Visualization of Raw Data ......................... 192  
5.10.3 Mirrors ........................................... 193  
5.10.4 Clipping Planes and Sectioning of Models .......... 194  
5.10.5 Input/Output ..................................... 197  
5.11 Rendering Control ................................... 202  
5.12 Conclusion .......................................... 202  
5.13 Future Work ......................................... 203

6 Case Study - The Visualization of Left-Ventricular Deformation 205  
6.1 Computing Ventricle Performance Measures ............. 206  
   6.1.1 Computing Volume Measures ........................ 206  
   6.1.2 Computing Ventricle Surface Areas ................. 208  
   6.1.3 Computing Length Measures ........................ 209  
6.2 The Visualization of Myocardial Strain .................. 209  
   6.2.1 Tensor Ellipsoids ............................... 209  
   6.2.2 Streamlines ..................................... 210  
   6.2.3 Hyperstreamlines ............................... 211  
   6.2.4 Line Integral Convolution ........................ 214  
   6.2.5 Colour Mapped Surfaces and Isosurfaces ............ 214  
6.3 Displacement Field Visualization ........................ 217  
6.4 Conclusion .......................................... 219  
6.5 Future Research ..................................... 219

7 Case Study - The Diffusion Tensor Field in the Brain 221  
7.1 Diffusion Measures ................................... 222  
7.2 Exploration of a Diffusion Tensor Data Set .............. 223  
   7.2.1 Colour Mapped Surfaces ........................... 223  
   7.2.2 Anatomical Landmarks ............................ 228  
   7.2.3 Introducing Local Tensor Information .............. 230  
   7.2.4 Extracting Nerve Fiber Tracts ..................... 236  
   7.2.5 Oblate Anisotropic Diffusion .................... 242  
   7.2.6 Influence of parameters .......................... 243  
7.3 Conclusion .......................................... 245  
7.4 Future Research ..................................... 246

8 Conclusion ............................................. 247  
8.1 Achievements & Contributions .......................... 247

9 Future Research ....................................... 251  
9.1 The Future of Medical Imaging .......................... 252  
9.2 Summary and Outlook .................................. 253

A Publications .......................................... 255
<table>
<thead>
<tr>
<th>Contents</th>
<th>ix</th>
</tr>
</thead>
<tbody>
<tr>
<td>B CD</td>
<td></td>
</tr>
<tr>
<td>C Mathematical Background</td>
<td></td>
</tr>
<tr>
<td>C.1 Polar Decomposition of a Symmetric 2D Second-Order Tensor</td>
<td></td>
</tr>
<tr>
<td>C.2 Cylindrical Coordinates</td>
<td></td>
</tr>
<tr>
<td>C.2.1 Coordinate Transformation of a Vector Quantity</td>
<td></td>
</tr>
<tr>
<td>D The Finite Element Method - Examples</td>
<td></td>
</tr>
<tr>
<td>D.1 2D Heat Conduction</td>
<td></td>
</tr>
<tr>
<td>D.2 Linear Elasticity</td>
<td></td>
</tr>
<tr>
<td>D.3 Divergence Theorem for a Symmetric Tensor</td>
<td></td>
</tr>
<tr>
<td>E Abbreviations</td>
<td></td>
</tr>
<tr>
<td>F Glossary</td>
<td></td>
</tr>
<tr>
<td>Bibliography</td>
<td></td>
</tr>
</tbody>
</table>
2.1 A rubber band before (a) and after (b) stretching .................. 10
2.2 A body before and after deformation .............................. 10
2.3 An infinitesimal quadratic area in the $xy$-plane has been strained without change of area (left). Rotating the element (right) shows that the deformation is equivalent to a translation in $x_1$-direction by an amount proportional to the $x_2$-coordinate followed by a rotation. The decrease in angle between the axes of the infinitesimal square element is approximately $2\varepsilon_{12}$. .......................... 12
2.4 Definition of a stress vector ........................................ 13
2.5 The Stress components $\sigma_{11}$, $\sigma_{21}$, and $\sigma_{31}$ are interpreted as components of a force acting on the surface orthogonal to $e_1$ (shaded) of an infinitesimal small axis-aligned unit cube .......................... 13
2.6 A slender cylinder of a linear-elastic material under a uniaxial load .......................... 14
2.7 Mathematical description of the 1D heat conduction problem ........... 18
2.8 Finite Element Approximation of the metal rod in figure 2.7 and the corresponding connectivity matrix .......................... 19
2.9 Linear Lagrange basis functions ................................. 20
2.10 Piecewise linear Lagrange interpolation (solid) of a quartic function (dashed) .......................... 20
2.11 Cubic Hermite basis functions ................................. 21
2.12 Piecewise cubic Hermite interpolation (solid) of a quartic polynomial (dashed) .......................... 21
2.13 The parent element for a 2D isoparametric quadrilateral .......................... 22
2.14 Bilinear element in world coordinates .......................... 22
2.15 Bilinear Lagrange basis functions ................................. 23
2.16 Bicubic Hermite element in world coordinates .......................... 25
3.1 Schematic drawing of a heart (a) with an example of a short axis (SA) and a long axis (LA) slice and a long axis (b) and short axis (c) tagged MRI image of the heart. All three images show the left ventricle (LV), the right ventricle (RV) and the endocardial surface and epicardial surface (in yellow) of the heart .......................... 33
List of Figures

3.2 Regions of the left-ventricular myocardium (the orientation is as in figure 3.1 (a)) ................................................. 33
3.3 Drawing of a longitudinal cross section of the heart (modified and annotated version of a figure from [Tex] ©2003 Texas Heart Institute) .................................................. 35
3.4 The normal electrocardiogram (ECG) (©1999 University of Tasmania, Department of Physiology [Unib]) ................................. 36
3.5 Atrial aspect of the myocardium with bundles of muscle fibers visible (©1980 Gower Medical Publishing [And80]) ......................... 37
3.6 Myocardial microstructure (©1993 CRC Press [HNS+93]) .......... 37
3.7 Tagged MRI images of a mid-ventricular short axis slice for a complete heart cycle (ordered from left to right and top to bottom). The first image shows the heart at end-diastole and the 10th image shows the heart at end-systole. The annulus shape in the centre of each image represents the left ventricular wall [The images were produced with our toolkit from tagged MRI data provided by Alistair A. Young] .......................... 46
3.8 Tagged MRI images of a mid-ventricular long axis slice for a complete heart cycle. The first image shows the heart at end-diastole and the 10th image shows the heart at end-systole. The left ventricular cavity is the semi-ellipsoidal gray shape on the right hand side of each image [The images were produced with our toolkit from tagged MRI data provided by Alistair A. Young] ................................. 47
3.9 For each image plane and each stack of tag planes one-dimensional displacement values Δx can be measured along each tag line .......................... 49
3.10 Tag lines and ventricular contours before (a) and after (b) myocardial contraction and the fitted epicardial and endocardial surfaces (c) [With kind permission from Dr. Alistair A. Young ©December 2002 ] .................................................. 54
3.11 The finite element model of the left ventricle at end-diastole .......... 55
3.12 The finite element model of the left ventricle at end-systole .......... 55
3.13 A wireframe representation of the finite element model of the dog heart during contraction. The element faces containing the ventricular cavities are rendered as blue surfaces and the myocardial fiber direction is indicated by red lines. The cavity on the left hand side of the image is the left ventricle .................................................................................. 57
3.14 Lateral view of the brain showing the cerebellum (4) and the cerebrum consisting of the frontal, parietal, occipital, and temporal lobe (the lobes are indicated by black lines and are listed in clockwise order starting from the left). The numbers 1,2,3,5, and 6 indicate the anterior, inferior, lateral, posterior, and the superior side, respectively (modified version of a figure from [EW91] ©1991 Wolfe Publishing Ltd.) .......................................................................................... 59
3.15 Sagittal section through the brain with the neuraxis indicated by two yellow line segments (the image was produced using a T1 weighted MRI data set and a volume visualization program obtained from [RSEB+00]) .................................................................................. 60
3.16 Horizontal section through the brain at the position of the neuraxis (the image was produced using a \(T_1\) weighted MRI data set and a volume visualization program obtained from \[RSEB^+00\]) .................. 61

3.17 Posterior view of a cast of the ventricles in the human brain. The structures numbered 2-5 form the lateral ventricles (©1991 Wolfe Publishing Ltd. [EW91]) .......................... 62

4.1 A visualization schema ........................................ 72

4.2 Images of the RGB (a), HSV (b) and CIE (c) colour spaces ........... 81

4.3 Density of matter of an astrophysical jet in intergalactic space visualized by using two different colour scales (©1993 IEEE Computer Society Press [KK93]) ......................... 85

4.4 Several examples of common colour scales and colour scales described in the literature .................................. 86

4.5 Direct volume rendering of a human pelvis data set. The images show muscle tissue (left image) and bone material (right image). Skin is represented as a thin semi-transparent layer (used with permission ©Peter Kulka 2001 [Kul01]) ...................... 91

4.6 Vector glyph representing the local field behaviour in a fluid flow (©1994 Academic Press [Pv94]) ......................... 96

4.7 Surface particles visualizing a turbulent flow (©1994 Academic Press [Pv94]) .................................................. 98

4.8 Magnetic field visualized using a texture generated by Gabor filters. Field orientation, strength, and potential are mapped to texture orientation, contrast and inverse size, and colour, respectively (©1995 Association for Computing Machinery [WK95]) ............................. 104

4.9 Vector field with a streamline through the pixel with the centre \(p_0\) (left), white noise texture (middle), and output texture of the pixel .................... 107

4.10 Top row: LIC with a kernel length of 40. From left to right: using white noise, using low pass filtered white noise, using low pass filtered white noise and contrast stretching the output texture. Bottom row: kernel length of 10, 20, and 160. All images are contrast stretched and use low-pass filtered white noise ....................... 107

4.11 Classification and example icons for critical points. \(R_1, R_2\) and \(I_1, I_2\) denote the real and imaginary parts of the eigenvalues of the Jacobian of the vector field, respectively (after [HH89]) .......... 110

4.12 Topology and critical point glyphs of the velocity vector field of a 3D flow on the surface of a hemisphere cylinder (©1991 IEEE Computer Society Press [HH91]) ........................................ 111

4.13 A topological surface visualizing the time history of a 2D flow past a circular cylinder (©1991 IEEE Computer Society Press [HH91]) ..................... 111

4.15 Hyperstreamlines and deformation surfaces visualizing the stress in a solid under a point load (indicated by the red arrow) (©1998 IEEE Computer Society Press [BP98b]) ........................................ 115
4.16 Trisector (left) and wedge point (right). Bold lines indicate separatrices and thin lines trajectories of the major eigenvalue field .... 118
4.17 A circular vector field visualized using different visualization icons . 123
4.18 Visualization of three 2D scalar fields and one vector field by using a height field, colour mapping, isocontours implemented by bump mapping, and vectors implemented by bump mapping (©1993 IEEE Computer Society Press [KK93]) ........................................ 126
4.19 A visualization of the nerve fiber structure in the brain .......... 129

5.1 An example of the visualization toolkit at work. The image shows the toolkit control at the top-left, two visualization windows partially covered in the middle, two visualization controls at the left, and one rendering control in the top-right and the colour map control partially covered in the bottom-right ........................................ 134
5.2 Top-level class diagram of the visualization toolkit ................. 136
5.3 Top-level class diagram of the field data structure ............... 138
5.4 Class diagram of (a subset of) the scalar field data structure ... 139
5.5 A visualization control (left) and the graphical user interface for creating a new field (right) .................................................. 141
5.6 The visualization of the field defined in figure 5.5 (left) and the user interface used to edit an expression field (right) .................. 142
5.7 Defining a macro and using it to create a new field ............... 142
5.8 A cube object with a translation handle (blue) and a modification handle (red). Part (b) of the figure shows the scene from part (a) after moving the blue handle which translates the cube object. Part (c) of the figure shows the deformed object obtained by moving the red handle .................................................. 144
5.9 An object is moved in 3D by converting a mouse movement on the screen to a movement in world coordinates \((x, y, z)\) parallel to the screen .................................................. 145
5.10 Manipulating an object in 3D is facilitated by splitting a window into four parts and by displaying simultaneously the current view (bottom-right) with projective views aligned with the three coordinate axes . 145
5.11 The visualization window in the top-middle of the figure shows a zoomed view of the heart model and the field probe with the material coordinate axes being displayed. The white lines indicate the finite elements of the model (see subsection 5.8.1). The top-right window is used to create the field probe and allows the user to define the information which is displayed in the “field probe output” window shown in the bottom-right of the figure .................. 147
5.12 (a) Subset of the colour spectra available in the toolkit. (b) User interface for creating piecewise linear colour scales .................. 149
5.13 Two versions of the piecewise linear temperature colour spectrum ........ 149
5.14 A colour mapped surface using Gouraud shading (a) and texture mapping (b) ................................................................. 151
5.15 Features of colour maps: (a) spectrum marker, (b) discrete colour map, (c) cyclical colour map, (d) exponential colour map .......... 152
5.16 Color maps generated from the temperature scale by reversing, inverting and exponentially stretching it: (a) the original temperature scale (b) the original scale reversed (c) the original scale exponentially stretched (d) the original scale exponentially stretched and reversed (e) the original scale exponentially stretched and inverted (f) the original scale exponentially stretched, reversed and inverted ........ 154
5.17 Diagram of the colour map class structure ............................... 155
5.18 Simplified class diagram of the data structure for selecting a volume-of-interest ................................................................. 156
5.19 User interface for creating volumes-of-interest in world coordinates . 157
5.20 Two examples of volumes-of-interest in world coordinates .......... 157
5.21 User interface for specifying element sets .............................. 158
5.22 Examples of element sets .................................................. 158
5.23 Simplified class diagram of the data structure for selecting a surface-of-interest ................................................................. 158
5.24 User interface for specifying surfaces-of-interest ...................... 159
5.25 Examples of various surfaces-of-interest .................................. 159
5.26 Simplified class diagram of the data structure for creating a point set over a region of interest ............................................. 160
5.27 User interface for creating a volumetric point set ....................... 161
5.28 Six examples of volumetric point sets ..................................... 161
5.29 User interface for creating a surface-based point set ................. 162
5.30 Examples of surface-based point sets .................................... 162
5.31 (a) Graphical user interface for filter selection. The illustration shows the tab for creating a simple range filter. (b) Example how a range filter is used to define a set of sample points with field values within the specified range. (c) A visualization of the direction of the principal stress using bidirectional vector glyphs located at regularly spaced sample points with a principal stress of greater than 140 (using the point set specified in (b)). The region specified by the filter is indicated by the red isosurface ........................................... 164
5.32 The user interface for defining a wireframe mesh (left) and the plate model rendered as a colour mapped wireframe (right) ...... 165
5.33 The user interface for computing the model surface (left), the interface for modifying the surface, and the resulting surface of the FE model (right) ................................................................. 166
5.34 The user interfaces for computing the volume of an element selection (left), the area of a surface (middle) and the arc length of a parameter curve (right) ......................................................... 169
5.35 A bulged plate with a hole (a) and the corresponding model of 1/4 of the plate under an uniaxial load $\mathbf{F}$ ........................................ 170
5.36 The user interface for creating a colour mapped surface (left), two colour mapped surfaces used to visualize the maximum principal stress in the plate with a hole (middle) and the colour mapped surface of the plate model with transparencies enabled (right) ................. 171
5.37 Triangulated cubes ............................................ 173
5.38 Isurface within a degenerate finite element approximated with one Marching Cubes cell ........................................ 174
5.39 A hole in the polygonization because of a face ambiguity .......... 174
5.40 The graphical user interface for creating an isosurface (left), the maximum principal stress visualized using four isosurfaces (middle) and the same visualization after applying the “sculpturing” tool (right) 175
5.41 The user interface for creating a height field (left) and the maximum principal stress on the bottom face of the plate with a hole visualized with a colour mapped height field (right) ......................... 176
5.42 The user interface for creating a particle system (left), the maximum principal stress visualized using volume-weighted randomly distributed particles (middle) and the regions of positive minimum principal stress (yellow-red) and maximum principal stress (green-blue) visualized using two particle systems with different particle sizes and colour maps (right) ........................................ 177
5.43 Isosurfaces can be identified by moving particles along the gradient of a scalar field until they reach the required isovalue ............... 178
5.44 The maximum principal diffusion in the brain visualized using a gray scale colour map ........................................ 180
5.45 The same data set visualized using a probabilistic distributed particle field .................................................. 180
5.46 The user interface for creating vector glyphs (left). A set of vector glyphs visualizing the displacement field in the plate with a hole under an uniaxial load (middle). The displacement field visualized by projecting a set of vector arrows onto a colour mapped surface (right) 180
5.47 (a) The displacement field (yellow arrows) and the minimum principal stress visualized using bidirectional arrows. Part (b) of the figure shows the same fields but this time the minimum principal stress field is visualized using red and blue coloured cylinders to indicate positive and negative principal stresses, respectively ............... 182
5.48 The user interface for creating a streamline bundle (left) and the results of using different integration methods (right). The yellow streamlines were computed with the Euler method and the red and green streamlines, which are virtually indistinguishable, were computed with the mid-point method and a 4th order Runge-Kutta method, respectively ........................................ 184
5.49 The user interface for creating a line integral convolution texture (left) and a visualization of the minor principal stress over a material plane of the plate with a hole (right) ........................................ 185
5.50 One octagonal section (red) of a unit sphere (yellow) .................. 187
5.51 Tensor ellipsoids visualizing the principal stress in the plate with a hole .......................................................... 187
5.52 The stress tensor visualized using hyperstreamlines in the direction of the maximum principal stress ................................. 188
5.53 Topology of a strain field with the maximum principal strain visualized by colour mapped spot noise .............................. 189
5.54 Example of markers used to highlight model features: The yellow sphere depicts the point of the maximum stress concentration, the blue rectangle the side of the model under uniaxial load and the green arrow indicates the load direction. The image shows additionally a red B-spline surface which is meaningless in this context but can be useful in anatomical models with more irregular features ........ 191
5.55 User interface for defining a label (left) and a visualization with two labels shown before (middle) and after (right) rotating the scene .. 192
5.56 The left ventricular FE model of a healthy heart (shown as a wire-frame) and one horizontal short axis MRI image from a stack of 8 horizontal images used to construct the FE model .......................... 193
5.57 Inserting mirrors into a scene makes it possible to observe different sides of a model simultaneously ................................. 195
5.58 The region of maximum principal stress in the plate with a hole is made visible by intersecting the model with three clipping planes . 196
5.59 A view inside a model is achieved by intersecting it with the union of three clipping planes ................................................. 196
5.60 The maximum principal strain in a plate with a hole visualized using colour mapped particles and by dividing the model into four sections 197
5.61 Several examples of models described using a meta-format ........... 198
5.62 A screen shot of the visualization of the diffusion tensor field in the brain viewed with a Web3D viewer over the internet ................. 200
5.63 A screen shot of the visualization of the mean diffusivity in the brain using a public domain volume visualization software ................ 201
6.1 Left ventricular cavity of the healthy heart at end-diastole (left) and end-systole (right) ..................................................... 207
6.2 Left ventricular cavity of the sick heart at end-diastole (left) and end-systole (right) ......................................................... 207
6.3 The strain field in the midwall of the healthy (left) and the diseased (right) left ventricle visualized using tensor ellipsoids. The septal wall is indicated by a yellow sphere ........................................ 210
6.4 The myocardial strain field in the healthy (left) and the diseased (right) left ventricle visualized using streamlines in the direction of the major principal strain. The septal wall is indicated by a yellow sphere .................................................. 211
6.5 The strain field in the healthy (left) and the diseased (right) left ventricle visualized using hyperstreamlines in the direction of the major principal strain. The septal wall is indicated by a yellow sphere .... 212
6.6 The strain field in the healthy (left) and the diseased (right) left ventricle visualized using hyperstreamlines in the direction of the minor principal strain. The septal wall is indicated by a yellow sphere .... 212
6.7 The strain field in the healthy (left) and the diseased (right) left ventricle visualized using hyperstreamlines in the direction of the minor principal strain. Perception of the complex 3D geometry is improved by rendering the endocardial wall in gray and by inserting mirrors into the scene. The septal wall is indicated by a yellow sphere .... 213
6.8 The minor principal strain (maximum contracting strain) in the healthy (top) and the sick (middle) heart visualized using Line Integral Convolution. The bottom images show the lateral wall of the healthy (left) and the sick (right) heart. The magenta coloured disks and lines indicate degenerate points ........................................ 215
6.9 The normal strain in the circumferential (top), longitudinal (middle) and radial (bottom) direction on the endocardial surface of the healthy (left) and the sick (right) heart visualized using colour mapping. The images show also the 0-isosurface which separates regions of contractile and expanding strain. The septal wall is indicated by a yellow sphere .................................................. 216
6.10 The circumferential-longitudinal shear strain on the endocardial surface of the healthy (left) and the sick (right) heart visualized using colour mapping. The images show additionally the 0-isosurface of this strain component. The septal wall is indicated by a yellow sphere .... 218
6.11 The displacement field of the contracting left ventricle visualized using vector arrows ........................................ 218
6.12 The displacement field of the contracting left ventricle visualized using vector arrows projected onto a radial-circumferential material plane ........................................ 218
7.1 Horizontal slices (with numbers) colour mapped with the maximum diffusivity ........................................ 224
7.2 Horizontal slices (with numbers) colour mapped with the mean diffusivity ........................................ 224
7.3 Horizontal slices (with numbers) colour mapped with the diffusion anisotropy ........................................ 224
7.4 Horizontal slice (number 20) with the mean diffusivity (a) and the diffusion anisotropy (b) visualized using an exponential (left) and a cyclical colour map (right) ........................................ 226
7.5 Horizontal slice (number 20) coloured using a segmentation function (a) and a barycentric colour map (b). In (a) red, green and blue indicate white matter, CSF and grey matter, respectively .......... 227

7.6 Horizontal slice (number 20) colour mapped with the oblate anisotropy (a) and the circular oblate anisotropy (b) ................. 228

7.7 The eye balls (green) and the ventricles of the brain (red) represented as isosurfaces of the mean diffusivity ..................... 229

7.8 Saggital section colour mapped with the mean diffusivity and overlaid with diffusion ellipsoids at the DTI grid vertices: 1-genu of corpus callosum, 2-fornix, 3-splenium of corpus callosum, 4-lateral ventricle, 5-fourth ventricle ................................................................. 230

7.9 A spherical colour map with hue, saturation and brightness varying along the circumferential, longitudinal and radial direction of the sphere, respectively. The illustration shows the surfaces of the colour map formed by choosing a constant brightness parameter of 1.0 (left) and 0.5 (right) ................................................................. 232

7.10 Fiber tract direction over a horizontal slice through the brain visualized using an ellipsoid-based texture. The ellipsoids are colour mapped with the spherical colour map in figure 7.9 and illuminated using ambient, diffuse and specular illumination (top), ambient and diffuse illumination (middle) and ambient illumination only (bottom). The images on the right show an enlargement of the region containing the splenium of the corpus callosum ......................... 233

7.11 Fiber tract direction over a horizontal slice through the brain visualized using an ellipsoid-based texture and the barycentric colour map shown in the bottom-left of the figure. The image in the top-left was rendered with ambient illumination only, the one in the top-right was rendered using ambient, diffuse and specular illumination .......... 234

7.12 Fiber tract direction over a horizontal slice through the brain visualized using anisotropy modulated line integral convolution (AMLIC). The nerve fibers in the splenium of the corpus callosum are parallel to the image plane (1) whereas nerve fibers in the posterior limb of the internal capsule (2) are almost vertical to it .............. 236

7.13 Fiber tracts visualized using streamtubes colour mapped with the diffusion anisotropy. The image shows from left to right and top to bottom the superior, posterior, anterior, inferior and the left lateral side of the brain ................................................................. 237

7.14 Nerve fiber tracts visualized by hyperstreamlines colour mapped with the maximum diffusivity: 1-Corona radiata, 2-Corpus callosum, 3-Optic radiation, 4-Internal capsule, 5-Cerebral peduncles, 6-Superior longitudinal fasciculus, 7-Splenium of the corpus callosum, 8-Inferior occipito-frontal fasciculus. The image shows the left lateral (a), posterior (b), superior (c) and anterior (d) side of the brain and a close-up view of the left lateral-posterior side (e) ...................... 239
7.15 Close up views of a visualization of fiber tracts using hyperstreamlines colour mapped with the diffusion anisotropy: 1-cingulum, 2-fornix, 3-middle cerebellar peduncle, 4-posterior limb of the internal capsule, 5-splenium of the corpus callosum....................... 241

7.16 Hyperstreamlines in the minor eigenvector direction visualizing the oblate anisotropy (a) and the circular oblate anisotropy (b) ........ 242

7.17 Hyperstreamlines generated using the parameters in table 7.1....... 244

C.1 A point $P$ in world coordinates $(x, y, z)$ and cylindrical coordinates $(r, \theta, z)$ together with its unitary basis $\{\hat{r}, \hat{\theta}, \hat{z}\}$ ......................... 264

C.2 A vector $\mathbf{v}$ in world coordinates $(x, y, z)$ and cylindrical coordinates $(r, \theta, z)$ ................................................................. 265

D.1 Mathematical description of the 2D heat conduction problem .... 268

F.1 Prolate spheroidal coordinates (©2002 Eric W. Weisstein [Wei]) ... 298
List of Tables

4.1 A classification of common visual attributes ...................... 84
4.2 Scalar icons .................................................. 120
4.3 Vector icons .................................................... 121
4.4 Tensor icons .................................................... 121
5.1 Myocardial volume of the healthy left ventricle at end-diastole and end-systole computed using different integration methods .......... 169
6.1 Myocardial volume (in $cm^3$) of the healthy and the diseased left ventricle at end-diastole (ED) and end-systole (ES) ............... 206
6.2 Ventricular volume (in $cm^3$) of the healthy and the diseased left ventricle at end-diastole (ED) and end-systole (ES), stroke volume (SV), and ejection fraction (EF) ...................................... 208
6.3 Surface area (in $cm^2$) of the endocardial and the epicardial surface of the healthy and the diseased left ventricle at end-diastole (ED) and end-systole (ES) ......................................... 208
7.1 Parameter choices for the hyperstreamline images in figure 7.17 .... 245
List of Tables

4.1 A classification of common visual attributes ........................................ 84
4.2 Scalar icons ..................................................................................... 120
4.3 Vector icons ................................................................................... 121
4.4 Tensor icons ................................................................................... 121

5.1 Myocardial volume of the healthy left ventricle at end-diastole and end-systole computed using different integration methods ........ 169

6.1 Myocardial volume (in cm$^3$) of the healthy and the diseased left ventricle at end-diastole (ED) and end-systole (ES) .......... 206
6.2 Ventricular volume (in cm$^3$) of the healthy and the diseased left ventricle at end-diastole (ED) and end-systole (ES), stroke volume (SV), and ejection fraction (EF) ........................................... 208
6.3 Surface area (in cm$^2$) of the endocardial and the epicardial surface of the healthy and the diseased left ventricle at end-diastole (ED) and end-systole (ES) .................................................... 208

7.1 Parameter choices for the hyperstreamline images in figure 7.17 ... 245
CHAPTER 1

Introduction

1.1 Motivation

During the past 100 years medical imaging has advanced from Roentgen’s original discovery of x-rays to a variety of 3D imaging tools such as magnetic resonance imaging, computed tomography, positron emission tomography and ultrasonography. Consequently the available medical data sets now comprise a diverse range of measurements such as tissue densities, blood flow velocity, and material strain. The development of mathematical models for organs and body parts has further increased the range of available data.

The size and complexity of biomedical data sets makes it increasingly difficult to understand, compare, analyze and communicate the data. Visualization is an attempt to simplify these tasks according to the motto “An image says more than a thousand words”. Representing complex material properties, such as strains, as a single image improves the perception of features and patterns in the data, enables the recognition of relationships between different measures and facilitates the navigation through and interaction with complex and disparate sets of data.

The aim of this thesis is the development of a toolkit for visualizing biomedical data sets with a particular emphasis on second-order tensor fields which are a fundamental entity in engineering, physical sciences and biomedicine. Examples are stresses and strains in solids and viscous stresses and velocity gradients in fluid flows. Large amounts of tensor data are particularly difficult to interpret since an \( n \)-dimensional second-order tensor has the same complexity as an \( n \times n \) matrix. An example for a biomedical tensor field is the strain field in the heart which describes the deformation of the heart muscle. It has been reported that abnormalities in the myocardial strain are visible before first symptoms of a heart attack occur [GZM97]. The goal of recording and visualizing cardiac data sets is to recognize and predict heart diseases which remain the biggest killer in the western world [MYPF00]. Understanding the deformation behaviour of the heart represented by the myocardial strain constitutes a major step towards this goal.
1.2 Contributions

The research accomplished in this thesis contributes to the disciplines of scientific visualization and biomedicine.

Chapter 4 summarizes results from cognitive science and offers guidelines for creating effective visualizations by applying perceptual concepts. We extend the traditional pipeline model for visualizing data to include two additional stages that take place within the observer: *visual perception* by the visual system and *cognition* by the human brain. An essential part of this model are visual attributes which we classify according to representational accuracy, perceptual dimension and spatial requirements. From these measures we obtain values for the information content and information density of a visual attribute.

Chapter 4 also presents a survey and an extended classification of visualization icons. By extracting the main visual attributes used for information mapping of each icon an informal measure of the suitability of an icon for different visualization tasks is obtained. The classifications for visual attributes and visualization icons combined with additional guidelines proposed in this chapter provide the scientist with a useful tool for selecting appropriate techniques for a given visualization task.

Chapter 5 presents a toolkit developed for exploring complex biomedical data sets. The contributions of this part are threefold: we suggest a modular design which facilitates the comparison and exploration of multiple data sets and visualizations. We also introduce a novel field data structure which allows interactive creation of new fields and we present boolean filters as a universal visualization tool. Our design incorporates finite element data structures and allows the definition of tissue properties in material coordinates, enables the selection of important structural components of the modeled organ (such as the inside or outside surface of the heart) and facilitates the computation of performance measures. We also suggest several improvements to some common visualization icons, e.g., we introduce cyclical colour maps and we describe tensor ellipsoids which encode the sign of an eigenvalue.

Chapter 6 and 7 present two case studies of practical importance. In the first case study we apply numerical concepts and visualization techniques traditionally used in solid mechanics and computational fluid dynamics to visualize models of a healthy and a diseased human left ventricle. We obtain new insight into the mechanics of the healthy and the diseased left ventricle and we facilitate the understanding of the complex deformation of the heart muscle by creating novel visualizations. Previously recorded results published as statistical data are confirmed and represented in an effective visual form. We also suggest a new hypothesis explaining the pumping behaviour of a left ventricle diagnosed with dilated cardiomyopathy.

The second case study presents visualizations of Diffusion Tensor Imaging (DTI) data. We propose (concurrently with another research group) a new method to extract and visualize nerve fiber tracts in the brain by using streamline integration. The nerve fibers tracts are visualized by streamtubes with a constant diameter or by hyperstreamlines which encode additionally the cellular water diffusion transverse to the fiber tract direction. Three additional novel visualization techniques
are presented: Barycentric colour maps allow an integrated view of different types of diffusion anisotropy; ellipsoid-based textures indicate diffusion direction and allow differentiation of tissue types but suffer from a lack of visual continuity; Anisotropy Modulated Line Integral Convolution (AMLIC) creates an image segmented by tissue type where white matter regions incorporate a texture which indicates the 3D orientation of nerve fiber tracts. The quality of our exploration approach and new visualization techniques is demonstrated by identifying various anatomical structures and features in a diffusion tensor data set of a healthy brain.

1.3 Thesis Overview

This thesis is divided into three parts. The first part comprises the chapters 2–4 and gives background information necessary for the understanding of this thesis from the fields of continuum mechanics, anatomy and physiology, medical imaging, bioengineering and scientific visualization. The second part consists of chapter 5 and introduces a visualization toolkit specifically designed for biomedical models and data sets. The capabilities of this toolkit are demonstrated in the third part of this work which comprises chapter 6 and 7 and contains case studies of biomedical data sets of the heart and the brain, respectively.

Chapter 2 introduces the finite element method which is frequently used to create biomedical models. We concentrate on the finite element representation of objects, i.e., the description of the object geometry and associated data fields by sample values and interpolation functions. The application of finite element analysis, i.e., finding a numerical solution to a set of partial differential equations governing the behaviour of a model, is inconsequential for the visualization process and is instead explained in appendix D. An important aspect of finite element modelling in the context of this thesis is the concept of material coordinates which are inherent to the modeled object and deform with it. As part of this opening chapter we also introduce the notions of stress and strain which are examples of second-order tensors and occur frequently in biological tissues.

Chapter 3 introduces two biomedical structures, the heart and the brain, which are used in the case studies presented later in this thesis. For each organ we describe its anatomy and where relevant its functioning, followed by a description of the corresponding biomedical models and associated data sets. The chapter is important in the context of this thesis for several reasons: it improves the understanding of the subsequent case studies and it enables the reader to compare visualization results with the actual organ anatomy and physiology. Furthermore the chapter helps to motivate the design and it helps to identify the required functionality of a visualization environment for biomedical structures. Finally the chapter introduces two examples of biomedical tensor data whose visualization is one of the main objectives of this work.

Chapter 4 is the final chapter of the introductory part of this thesis and it reviews the current state of the art of scientific visualization with an emphasis on tensor field visualization. The chapter starts with an overview of challenges encoun-
tered when visualizing multidimensional data and presents a summary of perceptual issues relevant for the design of a visualization. The next section introduces data transformation as a tool to simplify, expand, or modify data in order to make it more suitable for the visualization process. It is followed by a survey of existing visualization techniques for scalar, vector, and tensor fields. Methods which are important in the context of this thesis are dealt with in more detail. We conclude with a classification of visualization algorithms and summarize issues relevant for combining multiple visualization techniques into an effective visualization of a complex data set.

Chapter 5 constitutes the second and main part of this thesis. It describes an integrated visualization environment which we developed to investigate tensor field visualization methods. Although just a prototype implementation, the environment is a powerful visualization toolkit that allows the user to visualize complex biomedical data sets. In addition the application makes it possible to investigate and compare existing and novel visualization concepts. The chapter commences with an introduction of the top-level design of the toolkit and a description of the novel field data structure employed. Subsequently we explain various tools for user interaction, colour map design and the placement of visualization icons. We conclude with a summary of implemented visualization algorithms and tools for improving interaction with and perception of the data.

The final part of this work presents case studies of two biomedical data sets. In chapter 6 we evaluate the deformation of a healthy and a diseased human left ventricle. We present and explain various visualizations of the model and introduce methods for the efficient computation of performance measures from the finite element representation. We conclude with a discussion of our results and mention avenues for future research.

In chapter 7 we visualize the brain anatomy using Diffusion Tensor Imaging (DTI) data. Our approach starts with slice images familiar to the medical specialist and progressively expands the dimension and abstraction level of the representation in order to provide new insight into the data. The quality of our exploration approach and new visualization techniques are demonstrated by identifying various anatomical structures of the healthy brain.
During the past decade physically based modelling has emerged as an important
new technique in biomedicine and computer science. An important subfield is the
modelling of elastic bodies as used, for example, in computer animation [PW96]
and surgical simulation [SBM+94, KGPG96, CDA99]. A mathematical description
of elastic bodies is given by the theory of elasticity, the study of the deformation
of a solid body under loading together with the resulting stresses and strains. The
resulting mathematical models can be solved numerically using the finite element
method (FEM).

This thesis explores the visualization of biomedical structures represented by
finite element models. Consequently we are mainly interested in the finite element
(FE) representation of objects, i.e., the description of their geometry and data fields
by sample values and interpolation functions. Also important in the context of
this thesis is the concept of material coordinates which are inherent to the modeled
object and deform with it.

This chapter first introduces mathematical notations and definitions used in this
thesis. Next we give a short introduction into strains, stresses and the theory of
elasticity suitable for the computer scientist without an engineering background.
Then follows an introduction to the FEM with a particular emphasis on the FE
approximation of objects. The mathematical equations underlying the FE model
and the employed solution procedures are summarized briefly and their application
is demonstrated by two examples in appendix D.

The concepts introduced in this chapter are incorporated in the design of our
visualization environment and are used in the case studies in chapter 6 and 7.
2.1 Notations and Definitions

Vectors are written in small bold letters and matrices in bold capital letters or small bold Greek letters. The components of a vector $\mathbf{u}$ are $u_i, \ldots, u_n$, or, if necessary to avoid confusion, $u_x, u_y, u_z$ (in three dimensions). The components of a matrix $\mathbf{M}$ are $m_{ij}$ ($i, j = 1, \ldots, n$) so that the matrix $\mathbf{M}$ can also be expressed as $(m_{ij})$.

If the basis vectors of a vector space are constant, i.e., they have a fixed length and direction, the basis is called Cartesian. If the basis vectors are additionally unit and orthogonal the basis system is called rectangular Cartesian or just Cartesian. Unless stated otherwise we use a rectangular Cartesian coordinate system with the basis vectors $\mathbf{e}_i$, $i = 1, \ldots, n$. Vectors are by default taken to be column vectors.

The vector $\mathbf{p}$ from the origin to an arbitrary point $\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ is given by

$$\mathbf{p} = \sum_{i=0}^{n} x_i \mathbf{e}_i$$

(2.1)

The absolute value is denoted by $|\cdot|$, the determinant of a matrix by $\det$ or $|\cdot|$, the Euclidean norm by $||\cdot||$ and the gradient operator by $\nabla = \left( \frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n} \right)^T$. Matrix transposition is indicated by $^T$. Hence $\mathbf{u}^T \mathbf{v}$ is the dot product of the vectors $\mathbf{u}$ and $\mathbf{v}$ and $\nabla^T$ is the divergence operator, i.e., $\nabla^T \mathbf{f} = \frac{\partial f_1}{\partial x_1} + \ldots + \frac{\partial f_n}{\partial x_n}$. Furthermore $\mathbf{uv}^T = \mathbf{W}$ defines the outer product of two vectors where $w_{ij} = u_i v_j$.

2.1.1 Tensors

A $k$-th rank tensor (or tensor of order $k$) in $n$-space is a set of $n^k$ quantities which obey certain rules of transformation when the coordinate axes are rotated [App98, Wei]. A scalar is a tensor of order zero and a vector is a tensor of order one.

This thesis discusses the visualization of second-order tensors which are linear transformations between vectors and are represented by matrices. In the following unless stated otherwise the term tensor refers to a second-order tensor. Examples for tensors are stresses and strains which are explained in the next sections.

An important property of an $n$-dimensional symmetric second-order tensor $\mathbf{T}$ is that there always exist $n$ eigenvalues $\lambda_i$ and $n$ mutually perpendicular eigenvectors $\mathbf{v}_i$ such that

$$\mathbf{T} \mathbf{v}_i = \lambda_i \mathbf{v}_i \quad i = 1, \ldots, n$$

(2.2)

The above equations can be rewritten as

$$(\mathbf{T} - \lambda_i \mathbf{I}) \mathbf{v}_i = 0 \quad i = 1, \ldots, n$$
2.1 Notations and Definitions

The resulting linear system of equations has non-trivial solutions if and only if the matrix \((T - \lambda I)\) is invertible. Hence the eigenvectors of the tensors are given by the roots of the characteristic polynomial

\[
\det(T - \lambda I) = 0
\]

The matrix representation of a tensor is dependent on the reference coordinate system used. Suppose \(\{e_1, e_2, e_3\}\) and \(\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}\) are unit vectors spanning two Cartesian coordinate systems (i.e., the unit vectors are orthonormal). The components of a tensor \(T\) with respect to \(\{e_1, e_2, e_3\}\) are expressed with respect to \(\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}\) by the transformation [LRK86]

\[
\hat{T} = Q^T T Q
\]

where \(Q = (q_{ij})\) is the coordinate transformation matrix defined through the equations

\[
\hat{e}_i = Q e_i, \quad i = 1, \ldots, 3
\]

Note that

\[
q_{ij} = q_{ij} e_i^T e_i = \sum_j q_{ij} e_i^T e_j = e_i^T Q e_j = e_i^T \hat{e}_j = \cos(e_i, \hat{e}_j)
\]

since \(e_i^T e_j = \delta_{ij}\).

Several coordinate system independent measures can be derived from a tensor. In three dimensions these measures include the tensor invariants [LRK86]:

\[
I_1 = \text{trace}(T) = T_{11} + T_{22} + T_{33}
\]

\[
I_2 = \begin{vmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{vmatrix} + \begin{vmatrix} T_{11} & T_{13} \\ T_{31} & T_{33} \end{vmatrix} + \begin{vmatrix} T_{22} & T_{23} \\ T_{32} & T_{33} \end{vmatrix}
\]

\[
I_3 = |T|
\]

The three tensor invariants are the coefficients of the cubic equation

\[
\lambda^3 + I_1 \lambda^2 + I_2 \lambda + I_3 = 0
\]

whose solutions are the eigenvalues of \(T\).

Note that the first tensor invariant can also be expressed as \(I_1 = \lambda_1 + \lambda_2 + \lambda_3\). For a proof choose the eigenvectors of the tensor as a reference coordinate system. Representing the tensor in the resulting coordinate system gives a matrix with zero off-diagonal elements and the eigenvalues as diagonal elements.

If we represent the tensor by an ellipsoid with the lengths and directions of its principal axes given by the eigenvalues and eigenvectors then the first tensor invariant is proportional to the average of the axis lengths, the third invariant is proportional to the ellipsoid volume and the second invariant can be interpreted as a surface measure [Ale00].
2.2 Curvilinear Coordinates

While many modeling problems are formulated in three-dimensional Cartesian space, some problems, especially those involving curved geometric bodies, are better posed in a non-Cartesian curvilinear coordinate system. This subsection introduces curvilinear coordinate systems and shows how to transform between them and Cartesian coordinates.

Given an \( n \)-dimensional rectangular Cartesian space, a point \( p \) is given by a set of rectangular Cartesian coordinates \((r_1, \ldots, r_n)\). A new set of coordinates \((q_1, \ldots, q_n)\) can be defined by the transformation

\[
q_i = q_i(r_1, \ldots, r_n) \quad i = 1, \ldots, n
\]

If the Jacobian does not vanish, i.e.,

\[
|J| = \begin{vmatrix} \frac{\partial q_1}{\partial r_1} & \cdots & \frac{\partial q_1}{\partial r_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial q_n}{\partial r_1} & \cdots & \frac{\partial q_n}{\partial r_n} \end{vmatrix} \neq 0 \quad (i, j = 1, \ldots, n)
\]

an inverse transformation

\[
r_j = r_j(q_1, \ldots, q_n) \quad j = 1, \ldots, n
\]

does exist [Heu81, p. 300] with \(\left(\frac{\partial r_j}{\partial q_i}\right) = J^{-1}\).

If \( q_i \) is changed and \( q_j \) \((j \neq i)\) is held constant then the vector \( r \) will vary along a curve. This curve is called the coordinate curve for \( q_i \). When the above transformation is linear the new coordinate system is again Cartesian, though not necessarily orthogonal or normalized. When the above transformation is non-linear the new coordinate system is called curvilinear [Bat82]. An example of curvilinear coordinates is explained in detail in appendix C.2.

The position vector \( p \) can now be represented as a function of the new coordinates \( p = p(q_1, \ldots, q_n) \)

The partial derivatives

\[
\frac{\partial p}{\partial q_i} \equiv \hat{e}_i \quad i = 1, \ldots, n
\]

are tangent vectors of the coordinate curves \( q_i \). The vectors \( \hat{e}_i \) form a basis, the so-called unitary system, for the coordinates \((q_1, \ldots, q_n)\). In general the vectors are neither orthogonal nor normalized and using equation 2.1 can be expressed in terms of the Cartesian basis as

\[
\hat{e}_i = \frac{\partial p}{\partial q_i} = \sum_{j=1}^{n} \frac{\partial r_j}{\partial q_i} e_j \quad \text{(2.7)}
\]

Consequently a vector

\[
\hat{v} = \sum_{i=1}^{n} \hat{v}_i \hat{e}_i
\]
expressed in curvilinear coordinates can be written with respect to the original basis as

\[ \hat{v} = \sum_{i=1}^{n} \hat{v}_i \hat{e}_i = \sum_{i=1}^{n} \hat{v}_i \left( \sum_{j=1}^{n} \frac{\partial r_j}{\partial q_i} \hat{e}_j \right) = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial r_j}{\partial q_i} \hat{v}_i \hat{e}_j = \sum_{j=1}^{n} \tilde{v}_j \hat{e}_j \]

where

\[ \tilde{v}_j = \sum_{i=1}^{n} \frac{\partial r_j}{\partial q_i} \hat{v}_i \]

are the components of the vector in Cartesian coordinates. In matrix form this simplifies to

\[ \tilde{v} = J^{-1} \hat{v} \] (2.8)

An example of the representation of a vector with respect to different coordinate systems is given in subsection C.2.1.

Similarly a second-order tensor

\[ \hat{T} = \sum_{i=1}^{n} \sum_{j=1}^{n} \hat{t}_{ij} \hat{e}_i \hat{e}_j \]

expressed in curvilinear coordinates is written with respect to the Cartesian basis as

\[ T = \sum_{i=1}^{n} \sum_{j=1}^{n} \tilde{t}_{ij} \left( \sum_{k=1}^{n} \frac{\partial r_k}{\partial q_i} \hat{e}_k \right) \left( \sum_{l=1}^{n} \frac{\partial r_l}{\partial q_j} \hat{e}_l \right) = \sum_{i=1}^{n} \sum_{j=1}^{n} \tilde{t}_{kl} \hat{e}_k \hat{e}_l \]

where

\[ \tilde{t}_{kl} = \sum_{i=1}^{n} \sum_{j=1}^{n} \tilde{t}_{ij} \frac{\partial r_k}{\partial q_i} \frac{\partial r_l}{\partial q_j} \]

are the components of the tensor in Cartesian coordinates. In matrix form this simplifies to

\[ \tilde{T} = J^{-1} T (J^{-1})^T \] (2.9)

2.3 Linear Elasticity

2.3.1 Displacement and Strain

An elastic body under an applied load deforms into a new shape. The theory of linear elasticity provides a mathematical description for the displacement the body undergoes. For a one-dimensional example consider a thin rubber band as pictured in figure 2.1. Two arbitrary points \( P \) and \( Q \) are marked at the positions \( x \) and \( x + \delta x \), respectively. After deformation these points move to the positions \( x + u \) and \( x + u + \delta x + \delta u \), respectively, where \( u \) is called the displacement. The total length increase (displacement) between these points is \( \delta u \). The strain is now defined as the increase per unit length, i.e., it is the displacement gradient, which in one dimension is defined as

\[ \epsilon_x = \lim_{\delta x \to 0} \frac{\delta u}{\delta x} = \frac{du}{dx} \]
In higher dimensions the situation is more complicated. Figure 2.2 shows a body before and after deformation. Under deformation the points $P$ and $Q$ move to position $x' = x + u(x)$ and $x' + dx' = x + dx + u(x + dx)$, respectively, where $u$ is called the displacement field.

If the points are only an infinitesimal distance apart the vector between the deformed points

$$dx' = dx + u(x + dx) - u(x)$$

can be written as [LRK86]

$$dx' = dx + (\nabla u)dx$$
where the second-order tensor
\[ \nabla \mathbf{u} = \begin{pmatrix} \frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} & \frac{\partial u_1}{\partial x_3} \\ \frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} & \frac{\partial u_2}{\partial x_3} \\ \frac{\partial u_3}{\partial x_1} & \frac{\partial u_3}{\partial x_2} & \frac{\partial u_3}{\partial x_3} \end{pmatrix} \]
(2.10)
is known as the displacement gradient.

It can be seen that if \( \nabla \mathbf{u} = 0 \) then \( d\mathbf{x}' = d\mathbf{x} \) and the motion in the neighborhood of point \( P \) is that of a rigid body translation. The information about the material deformation around \( P \) is contained in \( \nabla \mathbf{u} \). It is desirable to define an entity which contains only information about deformation, but not about rotation. To do this consider two material vectors \( d\mathbf{x}_1 \) and \( d\mathbf{x}_2 \) issuing from point \( P \). Their dot product after transformation is ([LRK86])
\[ (d\mathbf{x}_1')^T d\mathbf{x}_2' = d\mathbf{x}_1^T d\mathbf{x}_2 + 2d\mathbf{x}_1^T \mathbf{E}^* d\mathbf{x}_2 \]
where the symmetric second-order tensor
\[ \mathbf{E}^* = \frac{1}{2} \left( (\nabla \mathbf{u}) + (\nabla \mathbf{u})^T + (\nabla \mathbf{u})^T(\nabla \mathbf{u}) \right) \]
is the Lagrangian strain tensor. Note that if \( \mathbf{E}^* = 0 \) the lengths and angles between the material vectors \( d\mathbf{x}_1 \) and \( d\mathbf{x}_2 \) remain unchanged, i.e., the deformation \( \nabla \mathbf{u} \) around point \( P \) is a rigid body transformation (i.e., rotation or translation). The components of \( \mathbf{E}^* \) are
\[ E_{ij}^* = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \sum_{k=1}^{3} \frac{\partial u_k}{\partial x_i} \frac{\partial u_k}{\partial x_j} \right) \]
For small deformations the displacement gradients \( \partial u_i/\partial x_j \) are small and the quadratic term of \( \mathbf{E}^* \) can be neglected giving the strain tensor \( \mathbf{e} \) with the components
\[ \epsilon_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \]
(2.11)
Lai et al. show [LRK86] that in this case \( \epsilon_{ii} \) can be interpreted as the unit elongation (increase per unit length) of a material element in the \( x_i \) direction. The components \( \epsilon_{ii} \) are called the normal strains. Furthermore the terms \( 2\epsilon_{ij}, i \neq j \) can be interpreted as the decrease in angle in radians between two material vectors initially in the \( x_i \) and \( x_j \) directions and are known as total shear strains. The components \( \epsilon_{ij} \) are known as average shear strains or just shear strains [eFu].

Figure 2.3 illustrates these concepts. In the left part of the figure an infinitesimal quadratic area in the \( xy \)-plane has been strained without change of area. The changes in length perpendicular to the given material direction are \( \partial u_2/\partial x_1 \) and \( \partial u_1/\partial x_2 \), respectively. By rotating the element as shown on the right hand side it can be seen that the deformation has been a simple shear, i.e., the deformation is equivalent to a translation in \( x_1 \)-direction by an amount proportional to the \( x_2 \)-coordinate followed by a rotation. The decrease in angle between the axes of the infinitesimal square element is approximately \( 2\epsilon_{12} \).
Figure 2.3. An infinitesimal quadratic area in the xy-plane has been strained without change of area (left). Rotating the element (right) shows that the deformation is equivalent to a translation in $x_1$-direction by an amount proportional to the $x_2$-coordinate followed by a rotation. The decrease in angle between the axes of the infinitesimal square element is approximately $2\epsilon_{12}$.

Note that by definition the strain tensor $\epsilon$ is symmetric so that equation 2.2 holds. The eigenvectors $v_1$, $v_2$, and $v_3$ of $\epsilon$ are the principal directions of the strain, i.e., the directions in which there is no shear strain. The eigenvalues $\lambda_1$, $\lambda_2$, and $\lambda_3$ are the principal strains and give the unit elongations in the principal directions. The maximum, medium, and minimum eigenvalue are called the maximum, medium, and minimum principal strain, respectively.

### 2.3.2 Stress

The previous subsection gave a purely kinematic description of the motion and deformation of an elastic body without considering the internal and external forces causing it. Internal forces are body forces\(^1\) acting throughout the body and external forces are surface forces acting on a real or imagined surface separating the body. The surface force at a point of the surface is described by a stress vector.

Consider a plane $S$ with normal $n$ through a point $P$ of the elastic body as shown in figure 2.4. Let $\Delta f$ be the force acting on a small area $\Delta A$ containing $P$. The stress vector $t_n$ in $P$ is defined as

$$t_n = \lim_{\Delta A \to 0} \frac{\Delta f}{\Delta A}$$

In classical continuum theory the resulting stress vector is the same for all surfaces through point $P$ with a tangent plane $S$ at $P$. It can be shown ([LRK86]) that

\(^1\)Body forces are forces that act on all particles in a body as a result of some external body or effect not in direct contact with the body under consideration. An example of this is the gravitational force exerted on a body. This type of force is defined as a force intensity per unit mass or per unit volume at a point in the continuum. Hence, when, for example, considering gravity the body force (per unit mass) is the gravitational acceleration $g$.\n
2.3 Linear Elasticity

independent of the choice of $S$

$$t_n = \sigma n$$

where the linear operator $\sigma$ is the stress tensor in $P$.

To interpret the components of the stress tensor $\sigma$ consider an infinitesimal small axis-aligned cube as shown in figure 2.5.

The stress tensor components $\sigma_{11}$, $\sigma_{21}$, and $\sigma_{31}$ are the components of the stress vector $t_{e_1}$, i.e., they are the components of the force acting on an infinitesimal surface orthogonal to $e_1$. The other components of $\sigma$ are interpreted similarly. The diagonal elements $\sigma_{11}$, $\sigma_{22}$, and $\sigma_{33}$ are called the normal stresses and the off-diagonal elements $\sigma_{12}$, $\sigma_{13}$, $\sigma_{23}$, $\sigma_{21}$, $\sigma_{31}$, and $\sigma_{32}$ are called the shear stresses. By using the conservation of angular momentum equation it can be shown [HB96] that $\sigma$ is in fact symmetric$^2$, i.e., $\sigma_{12} = \sigma_{21}$, $\sigma_{13} = \sigma_{31}$, $\sigma_{23} = \sigma_{32}$.

$^2$This is not the case if there are body moments per unit volume such as for a polarized anisotropic dielectric solid [LRK86].
In this thesis we concentrate on the visualization of strain tensors and diffusion tensors which are always symmetric. In section 5.9 we use as an example the stress field in a plate under an uniaxial load for which the resulting stress tensors are also symmetric.

As for the strain tensor the three eigenvectors of the symmetric stress tensor $\mathbf{\sigma}$ give the principal directions of the stress and the eigenvalues give the principal stresses. Each principal direction gives the normal direction of a plane on which the shear stresses are zero and the normal stress is the principal stress.

### 2.3.3 Model of a Linear Elastic Solid

Every continuum in motion must fulfill Newton’s laws of motion. For steady state solutions the continuum doesn’t experience acceleration and the second law of motion reduces to the equilibrium equations

$$
\sum_{j=1}^{n} \frac{\partial \sigma_{ij}}{\partial x_j} + \rho g_i = \sum_{j=1}^{n} \frac{\partial \sigma_{ij}}{\partial x_j} + f_i = 0 \quad i=1,\ldots,n \tag{2.12}
$$

where $\sigma$ is the stress tensor, $\mathbf{g}$ is the body force per unit mass, $\rho$ is the mass density and $n$ is the number of dimensions [Bur87]. The product of body force per unit mass and mass density is often referred to as internal load $\mathbf{f}$.

The above system of equations is valid for every continuum. However, in order to describe the response of a specific material under a specific loading additional material parameters are necessary.

Figure 2.6 shows a slender cylinder with length $l$, diameter $d$, and cross-sectional area $A$ under a uniaxial load $p$. If the length of the cylinder increases linearly with the applied load and the diameter decreases linearly with it the body is called a linear elastic solid.

![Figure 2.6. A slender cylinder of a linear-elastic material under a uniaxial load.](image)

In order to define material properties, material behaviour which is independent of the specimen size must be identified. Appropriate measures are the axial stress

$$
\sigma_a = \frac{p}{A}
$$

the axial strain

$$
\epsilon_a = \frac{\Delta l}{l}
$$
2.3 Linear Elasticity

and the lateral strain
\[ \epsilon_d = \frac{\Delta d}{d} \]

Two material coefficients can now be derived. The Young’s modulus (or, modulus of elasticity) defines the ratio of axial stress to axial strain under uniaxial loading
\[ E = \frac{\sigma_a}{\epsilon_a} \]

and the Poisson’s ratio defines the ratio of lateral strain to axial strain
\[ \nu = \frac{\epsilon_d}{\epsilon_a} \]

For some materials (e.g., fibrous materials) the Young’s modulus and Poisson’s ratio might depend on the orientation of the cylindrical test specimen with respect to the material microstructure. In this case the material is said to be anisotropic with respect to its elastic properties. Otherwise the material is isotropic with respect to its elastic properties. If the elastic properties are the same over the whole material it is said to be homogeneous, whereas if the elastic properties vary from one neighbourhood to another the material is said to be inhomogeneous.

For a linear elastic material the relationship between stress and strain is a linear one, i.e.,
\[ \sigma_{ij} = \sum_{k=1}^{n} \sum_{l=1}^{n} C_{ijkl} \epsilon_{kl} \]

or assuming small deformations
\[ \sigma_{ij} = \sum_{k=1}^{n} \sum_{l=1}^{n} C_{ijkl} \epsilon_{kl} \]  \hspace{1cm} (2.13)

where the fourth-order tensor \( C \) is known as the elasticity tensor.

Since \( \sigma \) and \( \epsilon \) are symmetric it can be shown that \( C \) contains 36 degrees of freedom, i.e., 36 material constants are necessary to completely describe the elasticity tensor [LRK86]. The stress-strain relationship simplifies considerably if the material is isotropic. In this case the elasticity tensor is isotropic, i.e., the tensor is invariant under orthogonal transformations (see glossary) and equation 2.13 can be shown to reduce to
\[ \sigma = \lambda(\epsilon_{11} + \epsilon_{22} + \epsilon_{33}) I + 2\mu \epsilon \]  \hspace{1cm} (2.14)

i.e., the stress-strain relationship is fully described by two material constants \( \lambda \) and \( \mu \) [LRK86]. The material constants, known as Lamé’s constants, can be expressed in terms of the previously introduced Young’s modulus and Poisson’s ratio as follows
\[ \lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)} \]
\[ \mu = \frac{E}{2(1 + \nu)} \]
Assuming an isotropic material equation 2.13 (Hooke’s law) simplifies to
\[
\begin{bmatrix}
\sigma_{11} \\
\sigma_{22} \\
\sigma_{33} \\
\sigma_{12} \\
\sigma_{23} \\
\sigma_{13}
\end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix}
1-\nu & \nu & \nu & 0 & 0 & 0 \\
\nu & 1-\nu & \nu & 0 & 0 & 0 \\
\nu & \nu & 1-\nu & 0 & 0 & 0 \\
0 & 0 & 0 & 1-2\nu & 0 & 0 \\
0 & 0 & 0 & 0 & 1-2\nu & 0 \\
0 & 0 & 0 & 0 & 0 & 1-2\nu
\end{bmatrix} \begin{bmatrix}
\epsilon_{11} \\
\epsilon_{22} \\
\epsilon_{33} \\
\epsilon_{12} \\
\epsilon_{23} \\
\epsilon_{13}
\end{bmatrix}
\]
or in matrix form
\[
\mathbf{\sigma} = \mathbf{C}\mathbf{\epsilon}
\]
where the stress and strain components are represented in vector form as
\[
\mathbf{\sigma}^T = (\sigma_{11} \quad \sigma_{22} \quad \sigma_{33} \quad \sigma_{12} \quad \sigma_{23} \quad \sigma_{13})
\]
and
\[
\mathbf{\epsilon}^T = (\epsilon_{11} \quad \epsilon_{22} \quad \epsilon_{33} \quad \epsilon_{12} \quad \epsilon_{23} \quad \epsilon_{13})
\]
respectively, and the matrix representation of the elasticity tensor is
\[
\mathbf{C} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix}
1-\nu & \nu & \nu & 0 & 0 & 0 \\
\nu & 1-\nu & \nu & 0 & 0 & 0 \\
\nu & \nu & 1-\nu & 0 & 0 & 0 \\
0 & 0 & 0 & 1-2\nu & 0 & 0 \\
0 & 0 & 0 & 0 & 1-2\nu & 0 \\
0 & 0 & 0 & 0 & 0 & 1-2\nu
\end{bmatrix}
\]

The finite element modeller implemented during the course of this thesis allows the user to simulate isotropic linear elastic materials under small deformations. Typical examples of such materials are metals and rubber. Even though biological materials are usually highly anisotropic and have a non-linear stress-strain relationship, the previously described idealized material properties have been used for biomedical applications such as real-time surgery simulation [BNC96] and the modelling of bones [Yet89, chapter 14].

### 2.4 The Finite Element Method

The Finite Element Method (FEM) became popular in the 1960’s as a tool to solve problems in structural mechanics numerically using computers. Solutions are obtained by numerically solving partial differential equations predicting the response of physical systems subjected to external influences [Bur87]. A solution is computed by subdividing the domain of the physical system into a finite element (FE) mesh and by approximating the governing differential equations by integral expressions over mesh elements.

Over the last two decades the FEM has become increasingly popular and is now an accepted tool in the fields of biomedicine (bioengineering) and computer graphics (modelling and animation). Applications include surgical simulation [SBM+94],
2.4 The Finite Element Method

This thesis is concerned with the visualization of finite element models of biomedical structures. Two such models are introduced in the next chapter. Since the visualization process only requires knowledge of the representation of the object geometry and the associated data fields this section emphasizes the FE approximation and interpolation rather than the methods of solving the equations. Another feature of the FE method important in this thesis is the definition of so-called material coordinates which reflect the geometry and/or physical properties of the modelled object and deform with it.

This section first introduces concepts used in the FEM. Then follows an introduction to the FE discretization and a description of various FE interpolation functions used to approximate the geometry and data fields of a model. The subsequent subsections introduce the concepts of world and material coordinates and contain an overview of numerical integration techniques used for the evaluation of FE integrals. We conclude this section with a short overview of the FE solution process. A more detailed explanation of the FE solution process using two examples is given in appendix D. An introduction to the FEM suitable for non-experts is found in [HP02, Bur87, HB96].

2.4.1 Concepts

Burnett introduces the following four concepts in any FE problem [Bur87]:

- The **system** is an object composed of various materials whose properties are described by material parameters.

- The **governing equations** define the behaviour of the system and consist usually of differential equations expressing conservation principles of some physical property or variational principles such as the minimization of a physical property. They may also include constitutive equations which contain physical properties of the materials that constitute the system. The free\(^3\) variable or variables in the governing equations represent the unknown FE solution. In the following we assume that governing equation contains just one free variable.

- The **domain** of the problem is the space over which the free variable is defined. Usually the domain is the region of space occupied by the system and/or the time interval over which the system changes its state.

- **Loading conditions** are externally originating physical quantities that interact with the system and cause its state to change. Loads acting in the interior of the domain are called **interior loads** and are part of the governing equations.

---

\(^3\)In Engineering a quantity which is varied in an experiment is often referred to as independent variable [ERC]. Chapter 4 will use a different definition of the term independent variable in conjunction with the visualization of multidimensional data sets. We use instead the terms free variable or unknown variable for the varied quantity.
Loads acting on the boundary of the domain are called **boundary loads** and form separate equations called **boundary conditions**.

The solution of the FE problem is a solution for the free variable which fulfills the governing equations and the boundary conditions over the domain.

Figure 2.7 explains the above terminology with an example: The system is a thin metal rod which is connected on the left side to a heat source with a constant temperature of $T = 30^\circ C$ and on the right side to an energy source with a heat flux (transfer of energy) of $q = 100 \text{W/m}^2$. The domain is the interval $[0, 2]$ on the $x$-axis. The governing equation is

$$-\frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) = 0$$

where $k = 10 \text{W/m}^{-1} \text{C}^{-1}$ is the thermal conductivity of the metal and the unknown or free variable is the temperature $T$. Finally the system has no internal loads and it has the boundary conditions

$$T = 30 \text{ at } x = 0$$

$$q = -k \frac{\partial T}{\partial x} = 100 \text{ at } x = 2$$

![Figure 2.7. Mathematical description of the 1D heat conduction problem.](image)

### 2.4.2 Finite Element Approximation

A finite element approximation involves discretizing the domain into a **finite element mesh** and representing the solution $u(x)$ for the free variable by values at the nodes of the mesh. The solution is extended over the entire domain by interpolating the nodal values of an element using **element basis functions**. Instead of defining element basis functions for each element it is convenient to define a **parent element** and to define a mapping between it and the element.

As an example consider the metal rod in figure 2.7. In order to use a common notation for the unknown we denote the temperature distribution with $u(x)$ and its finite element approximation with $\tilde{u}(x)$. Figure 2.8 shows the rod approximated by three elements $e_1, \ldots, e_3$ with the nodes $n_1, \ldots, n_4$, which have the nodal coordinates $x_1, \ldots, x_4$ and the nodal values $u_1, \ldots, u_4$, respectively. Each element $e_j$ has two local nodes $n_1^{(e_j)}$ and $n_2^{(e_j)}$ which are associated with their corresponding global node $n_i$.
by a connectivity matrix \( i = \Delta(j, k) \) where \( i \) is the global node index, \( j \) the element index, and \( k \) the local node index. Similarly we have local nodal coordinates \( x_k^{(e)} \) and local nodal values \( u_k^{(e)} \). Each element \( e_j \) is associated with a parent element defined by the \([0, 1]\) interval in the \( \xi \) parameter space, i.e., \( \xi(0) = x_1^{(e_j)} \) and \( \xi(1) = x_2^{(e_j)} \). Any point in the domain then has a (global) world coordinate \( x \) and an associated \( \xi \)-coordinate (material coordinate or element coordinate) \( \xi \). If the same basis functions are used for the geometry and the unknown variables the mapping \( x(\xi) \) from material to world coordinates is called isoparametric mapping.

### Linear Lagrange Basis Functions

A simple approximation \( \tilde{u}^{(e)} \) of the temperature distribution over the element \( e \) is obtained by connecting the element nodal values with line segments. Mathematically this is expressed as an interpolation of the nodal values using Linear Lagrange basis functions \( \phi_i \)

\[
\tilde{u}^{(e)}(\xi) = \sum_{i=1}^{n} u_i^{(e)} \phi_i(\xi) \tag{2.16}
\]

where \( n = 2 \) and

\[
\phi_1(\xi) = 1 - \xi \quad \phi_2(\xi) = \xi \tag{2.17}
\]

The world coordinates of the corresponding values are obtained analogously by interpolating the element nodal coordinates

\[
\tilde{x}^{(e)}(\xi) = \sum_{i=1}^{n} x_i^{(e)} \phi_i(\xi)
\]

The basis functions, displayed in figure 2.9, have the property that \( \phi_1(0) = 1 \) and \( \phi_2(0) = 0 \). Hence \( \tilde{u}^{(e)}(0) = u_1^{(e)} \) and \( \tilde{x}^{(e)}(0) = x_1^{(e)} \). Similarly \( \phi_1(1) = 0 \) and \( \phi_2(1) = 1 \) such that \( \tilde{u}^{(e)}(1) = u_2^{(e)} \) and \( \tilde{x}^{(e)}(1) = x_2^{(e)} \). As a result the interpolation is \( C^0 \) continuous across element boundaries if nodes are shared between neighbouring elements.
As an example consider the linear Lagrange interpolation of the function $u(x) = 16 - 27x + 18x^2 - 5x^3 + 0.5x^4$ with the nodal coordinates $x_i = i$, $i = 1, \ldots, 4$ and the nodal values $u_i = u(i)$. Figure 2.10 shows the original function as a dashed curve and the piecewise linear interpolation as a solid polyline.

Cubic Hermite Basis Functions

A smoother approximation than with the linear Lagrange interpolation is achieved by using a cubic Hermite interpolation. The interpolation requires in addition to the element nodal values $u_i$ and coordinates $x_i$ also the corresponding first derivatives $\left(\frac{du}{d\xi}\right)_i$ and $\left(\frac{dx}{d\xi}\right)_i$. If nodal values and derivatives are shared between elements the interpolation is $C^1$ continuous across element boundaries. The resulting interpolation function is

$$\tilde{u}^{(e)}(\xi) = \sum_{i=1}^{n} \left( u_i^{(e)} \phi_0^{(e)}(\xi) + \left( \frac{du}{d\xi} \right)_i^{(e)} \phi_1^{(e)}(\xi) \right)$$  \hspace{1cm} (2.18)

where $n = 2$. The element basis functions

$$\phi_0^0(\xi) = 1 - 3\xi^2 + 2\xi^3 \quad \phi_0^1(\xi) = \xi(\xi - 1)^2 \quad \phi_1^0(\xi) = \xi^2(3 - 2\xi) \quad \phi_1^1(\xi) = \xi^2(\xi - 1)$$  \hspace{1cm} (2.19)

are displayed in figure 2.11. The subscript gives the node index and the superscript indicates whether the basis function is associated with a nodal value (0) or a nodal derivative (1).

Note that the basis functions have the properties

$$\phi_i^j(0) = \begin{cases} 1 & \text{if } i = 1, \ j = 0 \\ 0 & \text{otherwise} \end{cases}, \quad \phi_i^j(1) = \begin{cases} 1 & \text{if } i = 2, \ j = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\left. \frac{d\phi_i^j}{d\xi} \right|_{\xi=0} = \begin{cases} 1 & \text{if } i = 1, \ j = 1 \\ 0 & \text{otherwise} \end{cases}, \quad \left. \frac{d\phi_i^j}{d\xi} \right|_{\xi=1} = \begin{cases} 1 & \text{if } i = 2, \ j = 1 \\ 0 & \text{otherwise} \end{cases}$$
and therefore both nodal values and nodal derivatives are interpolated, i.e.,

\[ \tilde{u}^{(e)}(0) = u_1^{(e)}, \quad \frac{d\tilde{u}}{d\xi} \bigg|_{\xi=0} = \left( \frac{du}{d\xi} \right)_1^{(e)}, \quad \tilde{u}^{(e)}(1) = u_2^{(e)} \]

\[ \frac{d\tilde{u}}{d\xi} \bigg|_{\xi=1} = \left( \frac{du}{d\xi} \right)_2^{(e)}. \]

![Figure 2.11. Cubic Hermite basis functions.](image)

![Figure 2.12. Piecewise cubic Hermite interpolation (solid) of a quartic polynomial (dashed).](image)

The smoother approximation of the cubic Hermite interpolation is demonstrated by taking again the function \( u(x) = 16 - 27x + 18x^2 - 5x^3 + 0.5x^4 \) as an example. As previously the coordinates are \( x_i = i, \ i = 1, \ldots, 4 \) and the nodal values are \( u_i = u(i) \). In addition the interpolation requires the nodal derivatives \( \left( \frac{du}{d\xi} \right)_i = \left. \frac{du}{dx} \right|_{x=x_i} \). Figure 2.12 shows the original function as a dashed curve and the interpolated function as a solid curve.

The definition of the cubic Hermite interpolation presented above requires that the nodal derivatives are given with respect to the \( \xi \)-coordinates. This is inconvenient in some applications since a shared node might need different nodal derivatives for each element in order to achieve \( C^1 \) continuity across neighboring elements.

As an example consider the function \( u(x) = x^2 \) over the element domains \( \Omega^{(e_1)} = [0, 1] \) and \( \Omega^{(e_2)} = [1, 3] \). The global node \( n_2 \) with the coordinate \( x_2 = 1 \) corresponds to the local nodes \( n_2^{(e_1)} \) and \( n_2^{(e_2)} \). Introducing an isoparametric mapping gives \( u^{(e_1)}(\xi) = \xi^2 \) and \( u^{(e_2)}(\xi) = (2\xi + 1)^2 \) where \( 0 \leq \xi \leq 1 \). The derivatives are \( \frac{du^{(e_1)}}{d\xi} = 2\xi \) and \( \frac{du^{(e_2)}}{d\xi} = 4(2\xi + 1) \) so that at the common node

\[ \left( \frac{du}{d\xi} \right)_2^{(e_1)} = 2 \neq 4 = \left( \frac{du}{d\xi} \right)_1^{(e_2)}. \]

A more useful formulation in practice is to define global nodal derivatives \( \left( \frac{du}{ds} \right)_I \), where \( s \) is an arc length parameter from the isoparametric mapping and \( I \) is the global node index. The corresponding element nodal derivatives are then computed as

\[ \left( \frac{du}{d\xi} \right)_i^{(e)} = \left( \frac{du}{ds} \right)_I \left( \frac{ds}{d\xi} \right)_i^{(e)}. \]
where \( \left( \frac{ds}{d\xi} \right)_i^{(e)} \) is an element scale factor which scales the arc length derivative of the global node \( I \) to the \( \xi \)-coordinate derivative of the element node \( i \).

The idea is demonstrated using the previous example. Using the arc length parameterizations \( s = \xi \) and \( s = 2\xi + 1 \) for the elements \( e_1 \) and \( e_2 \), respectively, yields the global nodal derivative \( \left( \frac{du}{ds} \right)_1 = 2 \) for the common node. The local nodal derivatives are computed as

\[
\left( \frac{du}{d\xi} \right)^{(e_1)}_2 = \left( \frac{du}{ds} \right)_2 \left( \frac{ds}{d\xi} \right)^{(e_1)}_2 = 2 \times 1 = 2 \\
\left( \frac{du}{d\xi} \right)^{(e_2)}_1 = \left( \frac{du}{ds} \right)_2 \left( \frac{ds}{d\xi} \right)^{(e_2)}_1 = 2 \times 2 = 4
\]

The finite element modeller incorporated into our visualization toolkit implements both finite elements with local and elements with global nodal derivatives.

### Multidimensional Basis Functions

Multidimensional finite elements are constructed as products of the underlying 1D elements. The domain of the parent element is formed from the product of the 1D parent element domains and the basis functions are defined as the tensor product of the corresponding 1D basis functions.

**Figure 2.13.** The parent element for a 2D isoparametric quadrilateral.

**Figure 2.14.** Bilinear element in world coordinates.

As an example consider a bilinear Lagrange element. The parent element is the unit square shown in figure 2.13. The \( n_{LL} = n_L n_L = 4 \) basis functions of the bilinear Lagrange element are the tensor products of the \( n_L = 2 \) basis functions of the linear
2.4 The Finite Element Method

Lagrange element given in equation 2.17, i.e.,

\[ \phi_1(\xi_1, \xi_2) = \phi_1(\xi_1)\phi_1(\xi_2) = (1 - \xi_1)(1 - \xi_2) \]
\[ \phi_2(\xi_1, \xi_2) = \phi_2(\xi_1)\phi_1(\xi_2) = \xi_1(1 - \xi_2) \]
\[ \phi_3(\xi_1, \xi_2) = \phi_1(\xi_1)\phi_2(\xi_2) = (1 - \xi_1)\xi_2 \]
\[ \phi_4(\xi_1, \xi_2) = \phi_2(\xi_1)\phi_2(\xi_2) = \xi_1\xi_2 \] (2.20)

and are shown in figure 2.15.

Figure 2.15. Bilinear Lagrange basis functions.

As in equation 2.16 the nodal values are interpolated as

\[ \tilde{u}^{(e)}(\xi) = \sum_{i=1}^{n} u_i^{(e)} \phi_i(\xi) \] (2.21)

where \( n = 4 \) and \( \xi = (\xi_1, \xi_2)^T \) are the material coordinates. The world coordinates are interpolated analogously, i.e.,

\[ \tilde{\mathbf{x}}^{(e)}(\xi) = \sum_{i=1}^{n} x_i^{(e)} \phi_i(\xi) \] (2.22)
where $\mathbf{x} = (x_1, x_2)^T$ and $\mathbf{x}_i^{(e)}$ are the 2D element nodal coordinates of the element $e$.

Figure 2.14 shows an example for a bilinear element in world coordinates.

Using the same procedure as for the bilinear Lagrange element it is also possible to define bicubic Hermite elements. The parent element is again the unit square shown in figure 2.13 and the $n_{HH} = n_H n_H = 16$ basis functions of the element are the tensor products of the $n_H = 4$ basis functions of the cubic Hermite element, i.e.,

$$
\phi_1^{0,0}(\xi_1, \xi_2) = \phi_1^0(\xi_1)\phi_2^0(\xi_2) = (1 - 3\xi_1^2 + 2\xi_2) (1 - 3\xi_1^2 + 2\xi_2^2)
$$

$$
\phi_1^{0,1}(\xi_1, \xi_2) = \phi_1^0(\xi_1)\phi_2^1(\xi_2) = \xi_1 (3 - 2\xi_1) (1 - 3\xi_1^2 + 2\xi_2^2)
$$

$$
\phi_2^{0,0}(\xi_1, \xi_2) = \phi_2^0(\xi_2)\phi_1^0(\xi_1) = (1 - 3\xi_2^2 + 2\xi_1) (3 - 2\xi_2)
$$

$$
\phi_2^{0,1}(\xi_1, \xi_2) = \phi_2^0(\xi_2)\phi_1^1(\xi_1) = \xi_2 (3 - 2\xi_2) \xi_2^2 (3 - 2\xi_2)
$$

$$
\phi_1^{1,1}(\xi_1, \xi_2) = \phi_1^1(\xi_1)\phi_2^1(\xi_2) = (1 - 3\xi_2^2 + 2\xi_1) \xi_2 (\xi_2 - 1)^2
$$

$$
\phi_2^{1,0}(\xi_1, \xi_2) = \phi_2^1(\xi_2)\phi_1^0(\xi_1) = \xi_1 (3 - 2\xi_1) \xi_1^2 (\xi_1 - 1)^2
$$

$$
\phi_1^{1,0}(\xi_1, \xi_2) = \phi_1^1(\xi_1)\phi_2^0(\xi_2) = \xi_1^2 (\xi_1 - 1) \xi_1^2 (3 - 2\xi_1)
$$

$$
\phi_2^{1,1}(\xi_1, \xi_2) = \phi_2^1(\xi_2)\phi_1^1(\xi_1) = \xi_1^2 (\xi_1 - 1)^2 \xi_2^2 (\xi_2 - 1)^2
$$

The nodal values are interpolated as

$$
\tilde{u}(\xi_1, \xi_2) = \sum_{i=1}^{n} \left( u_i^{(e)} \phi_i^{0,0}(\xi_1, \xi_2) + \left( \frac{\partial u}{\partial \xi_1} \right)_i^{(e)} \phi_i^{1,0}(\xi_1, \xi_2) + \left( \frac{\partial u}{\partial \xi_2} \right)_i^{(e)} \phi_i^{0,1}(\xi_1, \xi_2) + \left( \frac{\partial^2 u}{\partial \xi_1 \partial \xi_2} \right)_i^{(e)} \phi_i^{1,1}(\xi_1, \xi_2) \right)
$$

where $n = 4$. Note that in addition to the partial derivatives in the $\xi_1$ and $\xi_2$-directions at the element nodes the bicubic interpolation requires the mixed partial derivative $\left( \frac{\partial^2 u}{\partial \xi_1 \partial \xi_2} \right)_i^{(e)}$ at each node. An example of a bicubic element in world coordinates is shown in figure 2.16.

As in the 1D case it is again more practical to define global nodal derivatives $\left( \frac{\partial u}{\partial s_1} \right)_I$, $\left( \frac{\partial u}{\partial s_2} \right)_I$ and $\left( \frac{\partial^2 u}{\partial s_1 \partial s_2} \right)_I$, where $s_1$ and $s_2$ are the arc lengths in the $\xi_1$ and $\xi_2$ directions and $I$ is the global node index. The corresponding element nodal derivatives are then computed as

$$
\left( \frac{\partial u}{\partial \xi_1} \right)_i^{(e)} = \left( \frac{\partial u}{\partial s_1} \right)_I \left( \frac{ds_1}{d\xi_1} \right)_i^{(e)}
$$

$$
\left( \frac{\partial u}{\partial \xi_2} \right)_i^{(e)} = \left( \frac{\partial u}{\partial s_2} \right)_I \left( \frac{ds_2}{d\xi_2} \right)_i^{(e)}
$$

$$
\left( \frac{\partial^2 u}{\partial \xi_1 \partial \xi_2} \right)_i^{(e)} = \left( \frac{\partial^2 u}{\partial s_1 \partial s_2} \right)_I \left( \frac{ds_1}{d\xi_1} \right)_i \left( \frac{ds_2}{d\xi_2} \right)_i^{(e)}
$$
2.4 The Finite Element Method

2.4.3 Isoparametric Mapping

The previous subsection introduced the mapping of a unit square in \( \xi \)-coordinates to a finite element in world coordinates. The resulting isoparametric map was motivated by the goal to define the element basis function only once for a parent element. A further advantage of \( \xi \)-coordinates is that the coordinate system deforms with the material and as such allows representation of data values with respect to the deformed material. In many instances it is convenient to switch between world coordinates (\( x \)-coordinates) and material coordinates (\( \xi \)-coordinates). This subsection gives an overview of the mathematical principles involved. For simplicity we omit the element indices ‘(e)’ and the tilde-symbols indicating approximation functions.

Coordinate Transformation of a Point

Assuming a point in material coordinates \( \xi \) the corresponding world coordinates \( x \) are obtained by the isoparametric map

\[
x(\xi) = \sum_{i=1}^{n} \bar{x}_i \phi_i(\xi) \tag{2.23}
\]

where \( \bar{x}_i \) are the nodal coordinates. An example is the bilinear interpolation given by equation 2.22. Vice versa material coordinates can be computed from the world coordinates by using numerical approximation techniques such as a multidimensional Newton mesh [PVTF92].
Partial Derivatives with Respect to World and Material Coordinates

In most FE applications data quantities are defined by nodal values and are interpolated over the finite elements. For example, a scalar quantity \( u \) defined by the nodal values \( u_i \) is interpolated as

\[
u(\xi) = \sum_{i=1}^{n} u_i \phi_i(\xi)\]

The partial derivatives of \( u \) can be computed directly from the interpolation function as

\[
\frac{\partial u}{\partial \xi_j} = \sum_{i=1}^{n} u_i \frac{\partial \phi_i}{\partial \xi_j} \tag{2.24}
\]

The derivatives \( \frac{\partial \phi_i}{\partial \xi_j} \) of the basis functions are generally given as predefined analytic functions. The derivatives with respect to the world coordinate system are defined as

\[
\frac{\partial u}{\partial x_j} = \sum_{i=1}^{n} \frac{\partial u}{\partial \xi_i} \frac{\partial \xi_i}{\partial x_j}
\]

where \( \frac{\partial u}{\partial \xi_i} \) is defined as in equation 2.24 and the partial derivatives \( \frac{\partial \xi_i}{\partial x_j} \) are the components of the inverse of the Jacobian of the isoparametric mapping in equation 2.23

\[
J = \begin{pmatrix}
\frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} & \cdots & \frac{\partial x_1}{\partial \xi_n} \\
\frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} & \cdots & \frac{\partial x_2}{\partial \xi_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial x_n}{\partial \xi_1} & \frac{\partial x_n}{\partial \xi_2} & \cdots & \frac{\partial x_n}{\partial \xi_n}
\end{pmatrix}, \quad \begin{pmatrix}
\frac{\partial \xi_j}{\partial x_1} \\
\frac{\partial \xi_j}{\partial x_2} \\
\vdots \\
\frac{\partial \xi_j}{\partial x_n}
\end{pmatrix} = J^{-1} \tag{2.25}
\]

The introduced concepts can be generalised for vector and tensor quantities. Note, however, that a component based interpolation of vectors and tensors is not necessarily the most suitable one and alternative techniques have been suggested [AB99].

Coordinate Transformation of Vector and Tensor Quantities

Depending on the intended application it can be convenient to specify a quantity in either material or world coordinates. Material coordinates are often more computational efficient and provide a relationship between data values and underlying geometry. World coordinates are usually more convenient from a visualization standpoint. Whereas scalar quantities are coordinate system independent the representation of vector and tensor quantities depends on the coordinate system used.
Section 2.1 introduced the transformation of vector and tensor quantities between two coordinate systems. Using equation 2.8 the representations of a vector in material coordinates \( \mathbf{v} \) and in world coordinates \( \mathbf{\tilde{v}} \) are converted into each other by

\[
\mathbf{v} = \mathbf{J} \mathbf{\tilde{v}} , \quad \mathbf{\tilde{v}} = \mathbf{J}^{-1} \mathbf{v}
\]  

(2.26)

Similarly using equation 2.9 the corresponding formulas for converting tensors are

\[
\mathbf{T} = \mathbf{J} \mathbf{\tilde{T}} \mathbf{J}^T , \quad \mathbf{\tilde{T}} = \mathbf{J}^{-1} \mathbf{T} (\mathbf{J}^{-1})^T
\]  

(2.27)

Note that if the determinant of the Jacobian is zero, i.e., if the finite element is degenerate (e.g., two vertices have the same coordinates), the reverse transformations do not exist.

### 2.4.4 Gaussian Quadrature

Solving a FE problem requires the solution of complex integral equations involving the finite element basis functions (see appendix D). Similar integrals are encountered when computing model properties such as volume and surface areas (see subsection 5.8.2). Our visualization toolkit was specifically designed for FE models of biomedical structures and the computation and visualization of such derived information is a useful feature. An efficient evaluation of FE integrals is therefore desirable.

A popular method to evaluate integrals arising in 2D and 3D finite element problems is the Gaussian quadrature (Gauss-Legendre quadrature). The idea behind it is to approximate the integral of a function \( f \) over a range \([0,1]\) by weighted samples of \( f(\xi) \) taken at points \( \xi_1, \ldots, \xi_n \). If \( f \) is a polynomial of degree \( 2n - 1 \) then it has \( 2n \) coefficients. It is possible to choose \( n \) weights \( w_i \) and \( n \) gauss points \( \xi_i \) such that the approximation is exact [Bur87], i.e.,

\[
\int_0^1 f(\xi) \, d\xi = \sum_{i=1}^{n} w_i f(\xi_i)
\]

As an example take \( n = 2 \): We are looking for \( w_i, \xi_i, i = 1, 2 \), such that

\[
\int_0^1 f(\xi) \, d\xi = w_1 f(\xi_1) + w_2 f(\xi_2)
\]  

(2.28)

where \( f(\xi) = a_0 + a_1 \xi + a_2 \xi^2 + a_3 \xi^3 \). Expanding the left hand side of equation 2.28

\[
\int_0^1 f(\xi) \, d\xi = \sum_{i=0}^{3} a_i \int_0^1 \xi^i \, d\xi
\]

and applying equation 2.28 to each term yields a set of four equations

\[
\int_0^1 \xi^i \, d\xi = \frac{1}{i+1} = w_1 \xi_1^i + w_2 \xi_2^i, \quad i=0, \ldots, 3
\]
which is satisfied by
\[ w_{1,2} = \frac{1}{2}, \quad \xi_{1,2} = \frac{1}{2} \pm \frac{1}{2\sqrt{3}} \]

Similar calculations can be done to obtain the gauss points and weights for higher order polynomials [Bur87, Bat82, AS65]. The corresponding values for a 5th degree polynomial are [HP02]
\[ w_{1,3} = \frac{5}{18}, \quad \xi_{1,3} = \frac{1}{2} \pm \frac{1}{2}\sqrt{\frac{3}{5}} \]
\[ w_2 = \frac{4}{9}, \quad \xi_2 = \frac{1}{2} \]

The Gaussian integration is extended to multi-dimensional integration by defining gauss points whose components contain all combinations of the corresponding 1D gauss points with the corresponding weights multiplied (product rule). For example, choosing two gauss points in each coordinate direction results in the following 2D gauss points and weights
\[ w_{11} = w_{12} = w_{21} = w_{22} = \frac{1}{4} \]
\[ \xi_{11} = \left( \frac{1}{2} - \frac{1}{2\sqrt{3}} \right), \quad \xi_{12} = \left( \frac{1}{2} - \frac{1}{2\sqrt{3}} \right) \]
\[ \xi_{21} = \left( \frac{1}{2} + \frac{1}{2\sqrt{3}} \right), \quad \xi_{22} = \left( \frac{1}{2} + \frac{1}{2\sqrt{3}} \right) \]

The number of gauss points in each coordinate direction depends on the complexity of the element integrals to be evaluated. A complication in multi-dimensional problems is that the element integrals are often not polynomial due to the partial derivatives of the \( \xi \) parameter. Also the quadrature error has to be balanced against the discretization error. In general two gauss points are used in coordinate directions with quadratic basis functions and three gauss points in coordinate directions with cubic basis function [HP02]. Guidelines for choosing a suitable type of Gaussian integration are given by Burnett [Bur87] and Bahle [Bat82].

### 2.4.5 Solving a FE Problem

So far this chapter has described the finite element approximation of the model geometry and associated data fields. The actual solution process for a FE problem is not directly relevant in the context of this thesis since it usually does not influence the visualization process.

We have however implemented a complete FE modeller and employed it for the creation of various physical models used to test and demonstrate our visualization
2.4 The Finite Element Method

The following paragraphs give a short overview of the FE solution process.

A FE problem is solved by performing the following basic steps: First the domain is subdivided into smaller regions called finite elements. The shapes of the elements are limited to a certain type depending on the particular type of FEM chosen. Each type of element has a parent element and an associated set of basis functions (see subsection 2.4.2). Note that several different element types can be used simultaneously as long as they are compatible with each other.

The governing equation of the FE problem is then transformed into algebraic equations such that the equations are algebraically identical for each element of the same type. The terms of the element equations are numerically evaluated and assembled into an element stiffness matrix and an element load vector representing a linear system of equations (the so-called element equations or element system). The resulting matrices and vectors are assembled into one global stiffness matrix and global load vector which form a global system. The global system, which is a linear system of equations, is modified according to the imposed boundary conditions. The resulting equations describe the modeled system over the discretized given domain and are called the global system after imposing boundary conditions. The global system can be solved directly, e.g., by a Conjugate Gradient method [PVTF92], to yield the required solution.

For the interested reader the solution process is demonstrated by two examples in the appendix D. The first example describes a 2D heat conduction problem and was chosen for its simplicity. The second example describes a linear elastic solid under an applied load and is formulated in vector notation so that it can be immediately extended to higher dimensions. The 3D version of this example was employed to model a plate with a hole under an uniaxial load, which is used in chapter 5 for demonstrating various features of our visualization toolkit.

The enclosed CD contains an object-oriented FE modeller programmed by us using Microsoft Visual C++ 6.0 [Gre97]. The application implements generalised 2D and 3D versions of the FE problems discussed in appendix D. The graphical user interface was designed using OpenGL [WND97] and FLTK, a LGPL’d C++ graphical user interface toolkit for X (UNIX), OpenGL, and WIN32 [Spi].
CHAPTER 3

Biomedical Structures, Models and Data Sets

3.1 Introduction

This chapter introduces two important anatomical structures, i.e., the heart and the brain, which are visualized in the case studies presented in chapter 6 and 7, respectively. For each organ we describe its structure (anatomy), and where relevant its functioning (physiology), followed by a description of the corresponding biomedical finite element model and the associated data sets.

The chapter has two objectives. The main aim is to provide a basis for the subsequent case studies. We therefore summarize basic medical knowledge which makes it easier for the reader to compare visualization results with actual organ anatomy and physiology. We also introduce two tensor data sets used in the later case studies and we explain their medical significance. Of particular interest in a visualization context is the biomedical interpretation of that data, their relationship to organ diseases and abnormalities, and which visualization techniques researchers have employed so far for such data sets. The second objective is to motivate the design and the required functionalities of a visualization environment for biomedical structures.

The chapter is divided into two parts. The first part introduces the human heart and starts with a description of the structure and the functioning of the heart. An overview of two common heart diseases relevant to this thesis is followed by a survey of cardiac imaging methods with an emphasis on Magnetic Resonance Imaging (MRI), which was used to created the raw data used in the subsequent case studies. The following subsections explain the relevance of myocardial strain and stress in cardiac diagnosis, the computation of these measures and give an overview of previous visualization efforts. The section concludes with a description of the FE models of the heart used in this thesis.

The final section of this chapter introduces the anatomy of the human brain and
explains how Diffusion Tensor Imaging (DTI) can be used to obtain anatomical information. The section explains recent visualization work in this area and concludes with a description of the DTI data set used in the case study in chapter 7.

3.2 The Human Heart

The human heart is a hollow muscular organ weighting approximately $325 \pm 75g$ in men and $275 \pm 75g$ in women. It beats (contracts and expands) about 60-100 times a minute [And80, ASF+94] and pumps oxygen-poor blood to the lungs where red blood cells extract oxygen from air. The blood then flows back to the heart and is pumped through arteries to all parts of the body from where it flows back through the veins to the heart. The oxygen necessary for the working of muscles and other organs is extracted from the blood in capillaries, which are minute blood vessels connecting the smallest arteries to the smallest veins [ASF+94, JKR+01].

3.2.1 The Structure and the Functioning of the Heart

The heart has approximately the shape of an half-ellipsoid and contains two large chambers called the ventricles which are divided by the interventricular septum (see figure 3.1). The heart is situated inside the rib cage covered by the left and right lung with its long axis oriented from the right shoulder to the left upper abdominal quadrant. About two thirds of the heart is left of the mid-line of the body. The heart rests on the diaphragm, a muscular wall separating the thorax (chest) and the abdomen, with its apex tilted forward. The entire heart is contained in the pericardial sac which consists of an outer membrane (fibrous pericardium) and an inner membrane (serous pericardium). The sternal pericardial ligaments connect the fibrous pericardium to the sternum. These ligaments help hold the heart in place. The serous pericardium secretes a fluid into the pericardium to lubricate the heart when it is beating. Membrane plus fluid become a whole surface layer called the epicardium. The chambers of the heart are lined with a thin membrane called the endocardium. The heart muscle itself is referred to as the myocardium.

Myocardial Segmentation and Nomenclature

When discussing the heart it is convenient to introduce names for the different regions of the myocardium. Various types of segmentations and nomenclature for segments have been suggested. Often different conventions are used for different imaging modalities based on the strengths and weaknesses of each technique and the practical clinical applications. Recently an attempt has been made to standardize these options for all cardiac imaging modalities based on cardiac anatomy and clinical needs [CWD+02].

We use a segmentation and nomenclature described in [GKR+98]. Since the case study in chapter 6 examines the strain field in a left ventricle the following paragraphs present only terms describing the regions of the left ventricular myocardium.
3.2 The Human Heart

Figure 3.1. Schematic drawing of a heart (a) with an example of a short axis (SA) and a long axis (LA) slice and a long axis (b) and short axis (c) tagged MRI image of the heart. All three images show the left ventricle (LV), the right ventricle (RV) and the endocardial surface and epicardial surface (in yellow) of the heart.

As illustrated in figure 3.2 the myocardium is divided in circumferential direction into a septal, anterior, lateral, and inferior (or posterior) region. The anterior side of the left ventricle faces the chest, the inferior (posterior) side faces the back, and the septal region represents the interventricular septum. In the longitudinal direction the left ventricle is divided into an apical, a mid-ventricular or equatorial, and a basal region. Finally in the radial direction the myocardium is divided into a subendocardial, subepicardial, and midmyocardial region. The terms refer to the parts of the myocardium neighbouring the endocardium, the epicardium, and the region between them, respectively.

Figure 3.2. Regions of the left-ventricular myocardium (the orientation is as in figure 3.1 (a)).

Mixing these terms allows the description of 36 different regions of the left ven-
tricular myocardium, e.g., we can refer to the apical subendocardial septal region. Many authors (e.g., Moon et al. [MID\textsuperscript{+}94]) also employ a shorter alternative convention to refer to myocardial regions in the radial direction. For example, the lateral site of the septal wall can be called “septolateral” and the anterior site of the posterior wall can be referred to as “anterioposterior”.

In order to describe the position of other cardiac structures the following terms referring to the location of an anatomic structure with respect to the standing human body can be used: \textit{superior} means “towards the head”, \textit{inferior} means “towards the feet”, \textit{anterior} means “towards the abdominal surface of the body”, and \textit{posterior} means “towards the back of the body” [EW91].

A slice orthogonal to the apex-base axis of the heart is called a \textit{short axis} (SA) slice and a slice which contains the apex-base axis is called a \textit{long axis} (LA) slice (figure 3.1).

**Cardiac Structures and their Function**

The previous subsections introduced a simplified schematic description of the heart. Closer examination reveals that the heart contains four large blood vessels as illustrated in figure 3.3. The upper vessels (\textit{left and right atrium}) receive the blood and the lower vessels (\textit{left and right ventricle}) pump it out. The \textit{atrial septum} divides the upper vessels of the heart and the (inter)\textit{ventricular septum} divides the lower vessels. The left ventricle is the largest of the vessels and has a normal wall thickness of $9 - 11\text{ mm}$ during maximum expansion [SB99] with about 5\% higher values for athletes [FEP\textsuperscript{+}97]. The dimension of the left ventricle is $5.0 \pm 0.4\text{ cm}$ at maximum expansion and $3.1 \pm 0.4\text{ cm}$ at maximum contraction with values for males about 10\% higher than those of females [YICA94]. The right ventricular wall is approximately $3\text{ mm}$ thick [SB99] and is lined by muscle bundles, the \textit{trabeculae carnea} [ASF\textsuperscript{+}94, And80, KU, Kay].

The two sides of the heart perform different pumping actions. Oxygen-poor venous blood (dark, bluish-red) returns to the heart via the \textit{superior and inferior venae cavae} into the right atrium, where it is stored during right ventricular \textit{systole} (contraction). During \textit{diastole} (expansion) the blood flows from the right atrium to the right ventricle from where it is pumped out during systole through the main pulmonary artery (\textit{pulmonary trunk}) to the lungs. Simultaneously oxygen-rich blood (bright-red) flows from the lungs through pulmonary veins into the left atrium where it is stored during left ventricular systole. During diastole the blood flows into the left ventricle. Left atrial contraction provides a significant increment of blood to the left ventricle (“atrial kick”). During ventricular systole the blood is pumped through the aorta and its branches to all parts of the body.

The atria and the ventricles as well as the ventricles and their respective arteries are separated by valves which prevent the back flow of blood. The \textit{tricuspid valve} lies between the right atrium and the right ventricle and is composed of three cusps: the \textit{anterior cusp}, the \textit{posterior cusp}, and the \textit{septal cusp}. The \textit{mitral valve} lies between the left atrium and the left ventricle and consist of the anterior cusp and the posterior (mural) cusp. The \textit{pulmonary valve} separates the right ventricle and
the pulmonary trunc; and the aortic valve separates the left ventricle and the aorta. The latter two valves are composed of three semilunar cusps or leaflets: the right, left, and posterior cusp.

The papillary muscles control the cusps of the tricuspid valve and the mitral valve. They are contracted before the contraction of cardiac muscle. The cusps are connected to the papillary muscles by ligaments (chordae tendineae).

The heart itself is supplied with oxygen by the coronary arteries which originate from the right and left sides of the ascending aorta. The openings for the arteries are called the left and right coronary sinus and are just superior to the aortic valve. After being used by the heart muscles the oxygen-poor blood flows back to the heart via the coronary veins. The great cardiac vein and the middle cardiac vein empty into the coronary sinus which empties into the right atrium whereas the anterior cardiac vein empties directly into the wall of the right atrium.

### Electrical Activation and Motion Dynamics

The pumping action (beating) of the heart is coordinated by specialized tissue (neuratomyocardial cells) forming the heart’s own conduction system. Electrical impulses originate in the sino-atrial node (S-A node) located on the anterior surface of the superior vena cava (see figure 3.3). The cells of the S-A node spontaneously depolarise and thereby initiate an action potential, which is propagated rapidly through the atria which contract. The action potential then propagates slowly through the atrioventricular node (A-V node or junction) located on the right side of the intratrial septal wall to the ventricles (via the rapidly-conducting His-Purkinje system) which then also contract [ASF+94, Kay].

The potential of the electrical field originating in the heart can be measured on the body surface. The resulting graph is called an electrocardiogram (ECG or
EKG) and gives information about electrical conduction in the heart. The normal ECG has three distinguished features (see figure 3.4): the *P wave*, due to atrial depolarisation, is a moving wave with the positive charges in front of the negative charges. Usually the right atrium is activated before the left atrium which might cause a slight notch at the top of the wave. The *QRS complex* and the *T wave* are due to ventricular depolarisation and repolarisation, respectively. During repolarization the negative charges travel in front [Unib, ASF+94, Kay]. Recently images of the spatial-temporal distribution of the electrical potential in the myocardium have been obtained by employing optical mapping which uses a voltage-sensitive fluorescent dye [MSTM01].

The ECG is important in cardiac imaging since it can be used to identify the different stages of the heart cycle. For example, for the creation of the left ventricular model introduced in subsection 3.2.8 the image closest to the moment of maximum expansion (end-diastole) was determined by the rising R wave of the ECG.

Because the action potential does not spread instantaneously the heart muscle contracts with a rotating motion. During systole the base moves longitudinally towards the apex, which is essentially static [ACC+98], with the posterior wall moving farther than the anterior wall [YICA94]. The septum performs initially an anticlockwise rotation (apex-base view) but later a more radial movement. The apex rotates overall anticlockwise whereas the base rotates clockwise. The anteroseptal regions of the mid and apical levels and the posteroseptal region of the base perform a hooklike motion because of a reversal of rotation. The reversal is strongest in the posterior base which rotates initially anticlockwise but then clockwise by end-systole. Overall the septum rotates the least and the lateral and anterior walls the most with a higher rotation in the endocardial region than in the epicardial region [YICA94]. The base moves longitudinally towards the apex, which is essentially static [ACC+98], with the posterior wall moving farther than the anterior wall [YICA94]. In radial direction the lateral wall moves most and the septal wall moves least. The posterior wall moves more than the anterior wall at the apex but less than it at the base [YICA94].

The resulting torsion increases from base to the apex [ACC+98, BWR+90, YICA94] and is $4^\circ - 6^\circ$ higher in endocardial regions than in epicardial regions [YICA94]. The maximum twist in the left ventricle is $-0.06 \pm 0.02 rad/cm$ [MID+94]. In contrast
3.2 The Human Heart

to the rotation the torsion increases steadily during systole [YICA94] and reverses rapidly during isovolumic relaxation before diastolic filling [ACC+98, SSF+99]. The behaviour seems to be consistent with the time constant of isovolumic pressure decay \( \tau \), so that it might be suitable as a non-invasive measure of it [Rei99].

Reported values for the end-diastolic and end-systolic volume of the left ventricle are \( 96.4 \pm 34.5 \text{ml} \) and \( 47.2 \pm 30.5 \text{ml} \), respectively. The resulting stroke volume is \( 49.2 \pm 19.8 \text{ml} \) and the ejection fraction is \( 53.0 \pm 14.2\% \) [LSM+02].

In general the myocardium is considered incompressible but Denney and Prince estimate that small volume changes up to 10% occur due to myocardial perfusion [DP95]. This is consistent with results from Young et al. [YICA94] who report a net reduction of SA area during systole of approximately 15% at the apex, 10% at the midventricle, and 5% at the base, and a net reduction of LA area of between 14% at the apex and 4% at the base.

**Microstructure of the Heart**

Most of the myocardium is made up of contractile or “working” muscle cells (myocytes) which are about 50 – 100\( \mu \text{m} \) long and are about 10 – 20\( \mu \text{m} \) in diameter. Muscle cells have a long cylindrical shape and are arranged longitudinally in series; several of them forming a *muscle fiber* which in turn form fiber bundles (figure 3.5).

The axes of adjacent cells are parallel such that a fiber orientation can be defined as indicated in the top-left illustration of figure 3.6. Hunter et al. report [HNS+93] that the fiber orientation changes smoothly through the ventricular wall; in the left ventricular wall the angle of the muscle fibers with the circumferential direction is about \(-60^\circ\) at the epicardial surface, about \(0^\circ\) in the midwall, and about \(90^\circ\) at the endocardial surface.

![Figure 3.5. Atrial aspect of the myocardium](1980 Gower Medical Publishing [And80]).

![Figure 3.6. Myocardial microstructure with bundles of muscle fibers visible](©1993 CRC Press [HNS+93]).

Gower Medical Publishing [And80].
Parallel fibers form *fibril sheets* which are 3-4 layers thick. Fibers within a sheet are tightly coupled via intercalated disk junctions and a regular array of short collagen fibers. An intercalated disc connects the ends of two muscle cells. Fibers between sheets are coupled much less frequently by disk junctions and the collagen fibers connecting them are longer and more convoluted [HNS+93]. An illustration is given in the drawing at the bottom of figure 3.6. Large wavy collagen fibers traverse the heart wall in the approximate direction of the myocytes [HYL+99].

The mechanical properties of myocardial muscle fibers determine the deformation of the myocardium under load and therefore are important input parameters when simulating the contraction of the heart muscle. Until recently it was assumed that the myocardium is transversely isotropic, i.e., the properties of the muscle fiber are the same in all directions orthogonal to the fiber direction. Accordingly its mechanical properties were only tested under uniaxial load and it was determined that they obey a pole-zero law [HNS+93, Nas95]. However, it is known that during contraction the heart changes predominantly in diameter. LeGrice et al. [LTC95] reports 8% lateral expansion but 40% wall thickening. This indicates reorganization of the myocytes during systole. Because of the sheet structure of the myocardium it has been proposed that the sheets can slide over another restricted mainly by the length of the interconnecting collagen fibers [LTC95]. This assumption is supported by the fact that the fiber angle at the myocardial wall is higher during end-systole than during end-diastole [LTC95]. The shear properties of the myocardium resulting from this sliding motion are characterized in [DLS+00, DSYL02]. The shear is most restricted in the direction of the sheet normals and the maximum shear occurs in the fiber direction. Wall shear is thought to be an important mechanism of wall thickening during systole and therefore may play a substantial role in the ejection of blood from the ventricle [LTC95].

Recently several authors have employed Diffusion Tensor Imaging (DTI) in order to obtain an in vitro and/or in vivo measurement [SHS+01, MFE+01, ACCM01] of the myocardial fiber structure. A validation of the method is found in [HMM+98, SHWF98] while Masood and Yang review the technique and its use in combination with tagged MRI and myocardial velocity mapping [MYPF00].

### 3.2.2 Heart Diseases and Heart Failure

One or multiple heart diseases can result in heart failure, which is a clinical syndrome that arises when the heart is unable to pump sufficient blood to meet the metabolic needs of the body at normal filling pressures [ASF+94]. The goal of recording and visualizing cardiac data sets is to recognize and to predict heart diseases.

Causes of heart failure are differentiated into mechanical, myocardial, and rhythmic abnormalities [Kay, ASF+94]. Mechanical abnormalities include increased pressure or volume load (e.g., due to a dysfunctional valve) and bulging of the heart wall (*ventricular aneurysm*). Myocardial abnormalities include metabolic disorders (e.g., diabetes), inflammation, and *ischemia* (blockage of the coronary artery). Abnormalities of the cardiac rhythm or conduction disturbances include standstill, irregular
heart beat (fibrillation), and abnormally rapid heart beat (tachycardia). The following paragraphs concentrate on two diseases which are relevant in the context of this thesis.

**Atherosclerosis** is the narrowing of an artery due to a build-up of substances on the inner lining (intima) of the artery walls. It results in a reduced blood flow and hence decreased delivery of oxygen and other nutrients to the body tissues. The formation of a blood clot (thrombus) in the narrowed area can block the artery completely. Atherosclerosis in the coronary arteries causes **coronary artery occlusive disease (ischemic heart disease)**.

A **myocardial infarction** (heart attack) occurs when a coronary artery is completely blocked (stenosis) and an area of the heart muscle dies because it is completely deprived of oxygen for an extended period of time. Acute myocardial infarction starts in the subendocardium and spreads to the subepicardium within 20-40 minutes after occlusion of the coronary artery [LC99]. If the blood supply can be restored before the heart cells die, the patient will have a limited heart attack. Sometimes the body will do this on its own, by supplying blood through a system of alternate, un-blocked arteries (collateralization). Permanently damaged muscle is replaced by scar tissue, which does not contract like healthy heart tissue, and sometimes becomes very thin and bulges during each heart beat (aneurysm) [GZM97, ASF94].

Changes in myocardial perfusion as caused by ischemic heart disease lead to abnormalities in the heart’s deformation which is reflected in the myocardial strain field. Guttman et al. report that abnormalities in the myocardial strain are visible before first symptoms of a heart attack occur [GZM97] so that measuring and visualizing the strain might represent a useful diagnosis tool.

**Cardiomyopathy** refers to a variety a muscular disorders of the heart that are anatomically qualified by a thickening, thinning or stiffness of the whole or parts of the heart muscle. A particular type of this disorder is dilated cardiomyopathy, which is characterized by cardiac enlargement, increased cardiac volume, reduced ejection fraction, and congestive failure. The disease can be diagnosed from MRI images that show large left ventricular (LV) dilation and wall thinning, with right ventricular (RV) and atrial dilation sometimes also being present. MRI tagging shows additionally reduced cross-fiber shortening at the endocardium due to an underlying myocardial fibrosis and increased end-systolic wall stress [SB99]. Fiber and cross-fiber strain are reduced with a preserved transmural gradient [Rei99]. PET studies can further differentiate between ischemic cardiomyopathy in which there is an underlying perfusion deficit and idiopathic cardiomyopathy for which there is no known cause [Cru]. A case study of a heart with dilated cardiomyopathy is presented in chapter 6.

### 3.2.3 Cardiac Imaging

In order to assess the severity and treatability of a patient’s condition, a cardiologist might perform a series of diagnostic tests of increasing specificity, invasiveness, and
cost, as deemed necessary [GZM97]. The most basic tools are cardiac auscultation [Cab97], which involves listening to the sounds of the heart with a stethoscope, and electrocardiography, which records the electrical potential on the body surface by an electrocardiogram (ECG) [SAI+92, MJM92]. In many instances, however, these techniques are not sufficient for a diagnosis and 2D and 3D data sets of the heart must be created.

Various medical imaging modalities are in use today to create 2D and 3D data sets of the heart. A basic understanding of these techniques is necessary to create visualizations that appropriately reflect the relationship between data values and tissue properties and physiology. A powerful technique for creating 3D images of the heart is Magnetic Resonance Imaging (MRI) which was used to create the data sets of the left ventricle used in the case study in chapter 6. Before giving an introduction to MRI this subsection summarizes a couple of alternative cardiac imaging techniques that produce raw data suitable for our visualization environment.

Nuclear cardiology [ASF+94] involves injecting into the body a radioisotope which emits gamma rays (photons). A nuclear detector (gamma camera) registers gamma rays hitting it perpendicularly. In planar imaging a two-dimensional image is produced for each different view whereas single photon emission computed tomography (SPECT) requires data from multiple image planes forming a 180 degree arc to reconstruct a three-dimensional images. Special types of nuclear cardiology include myocardial perfusion imaging (MPI) where the deposition of an isotope in the myocardium is used to determine myocardial blood supply [Hen97b]; Cardiac blood pool imaging (BPI) which is used to measure volume changes in the heart chambers [Hen97a] and acute myocardial infarction imaging which uses radioactively labeled compounds that trace cardiac cells which break down [HLD+00].

Echocardiography (ECG) refers to the use of ultrasound to produce an image of the cardiac structure and function (echocardiogram). Ultrasound is reflected (echoed) on the interfaces between materials of different acoustic impedance. The time delay of the reflected signal determines the location of an anatomical structure whereas the amplitude of the reflected signal determines the brightness in the resulting image. Blood flow velocity can be recorded by using the Doppler principle [Duk]. ECG shows images in real-time but is limited by the size of the acoustic window and ultrasound penetration [SB99].

X-rays are electromagnetic radiation with a wavelength less than 1/10000 of that of visible light. A picture of body structures is obtained by passing x-rays through the body onto a film. Structures with a higher density (e.g., bones) absorb more x-rays and appear whiter on the exposed film. Soft tissue (such as the heart) is usually not well differentiated from the surrounding tissues since both hardly absorb any x-rays. A clearer picture can be obtained by injecting an x-ray absorbing contrast agent into the blood supply of the heart in order to show arteries, veins, and blood flow (coronary angiography). Cardiac fluoroscopy achieves the display of moving images by using a constant x-ray source and cesium iodine phosphors as an image intensifier [ASF+94].

X-ray computed tomography (CT) uses a rotating x-ray device (CT scanner) to
pass a series of x-ray beams through sections of a body. A computer registers those images and constructs a cross-sectional picture (slice) from them. With conventional CT the rotation of the x-ray tube is performed mechanically and requires 1–2s resulting in motion artifacts of moving organs such as the heart. Faster image acquisition is achieved using electron beam computed tomography (EBCT) which projects and detects multiple electron beams simultaneously [ASF+94].

Positron emission tomography (PET) [ASF+94, Cru] is similar to SPECT but uses isotopes that emit positrons. If a positron encounters an electron positron annihilation occurs and two 511 KeV photons are emitted with an angle of 180°. The photons can be detected and because of their simultaneous birth the exact location of the positron annihilation can be computed. In contrast to SPECT PET offers the opportunity to define absolute units of regional functional processes in the human heart such as blood flow, biochemical reaction rates, and neuronal activity. A common application in cardiac imaging is the measurement of glucose consumption, which, for example, can be used to detect viable myocardium that is likely to respond to revascularization [RdB99].

One of the most recent advances in medical imaging is molecular imaging which involves noninvasive mapping of cellular and sub-cellular molecular events [Wei99]. Several methods are used to facilitate this imaging, including PET, MRI, and optical imaging, which enables assessment of gene expression in vivo. Optical imaging uses either near-infrared fluorescence [WTMB99] or optical coherence tomography (see appendix F) and may provide better sensitivity and specificity at the molecular level than other modalities [Nat02].

Magnetic Resonance Imaging

Magnetic resonance imaging (MRI) uses a powerful cylindrical magnet (MRI scanner) to excite hydrogen atoms of a body placed inside it. The atoms give off radio-frequency waves which are translated into a visual image. The following paragraph summarizes the underlying physical principles and indicates how they are exploited to detect different tissue types.

Hydrogen atoms have the property that the nucleus spins and hence, like a current flowing through a wire, produces a magnetic field. When a body is placed into a magnetic field $B_0$ the nuclei orient themselves parallel to the field and produce a slight magnetic field themselves. By applying a second magnetic field $B_1$ rotating perpendicular to $B_0$ the spin of the nuclei is displaced from alignment with $B_0$. If the $B_1$ field is discontinued the nuclei continue to rotate and emit an radio-frequency signal that is recorded. The signal decays slowly due to other structures surrounding the nuclei ($T_1$ relaxation) and due to neighboring nuclei ($T_2$ relaxation).

Different tissue types have different $T_1$ and $T_2$ relaxation times. A variety of MRI techniques are obtained by exploiting this property and by applying different radio-frequency pulse sequences for the $B_1$ field in order to highlight static tissues or moving blood in the produced images or to determine blood dynamics. Many individual tissue properties of the MRI signal, such as MR-proton density, relaxation rates, flow, chemical shift, diffusion, and perfusion, contribute to soft tissue contrast.
Unlike CT, MRI can create an image of tissue slices from any direction or plane without moving the patient and allows greater contrast in soft tissues [Lin03]. Recent advances have considerably extended the possibilities of MRI and now include high-quality angiography, functional imaging, flow measurements, MR cholangiopancreatography [MS00] and the visualization of functional and metabolic processes (MRI spectroscopy) [SB99]. Using phase-contrast MRI blood velocity can be measured in different parts of the heart. Similar to EBCT analysis programs can be used to measure ejection fraction and cardiac volumes, size of vascular structures, myocardial mass, and ventricular wall thickness [ASF+94]. Myocardial deformation and strain can be measured using tagged MRI [ZPR+88, AD89] in which a non-invasive pattern of tissue-spin polarization changes is placed and tracked in the myocardium (see subsection 3.2.5).

A good introduction into cardiac MRI is found in [Box99, KPF+99] and comprehensive overviews of cardiovascular MRI are given in [FHdS99, SB99, Rei99, FG99]. Good reviews of cardiac tagged MRI are found in [McV96, MO01].

### 3.2.4 Myocardial Strain and Stress as an Indicator of Heart Failure

The analysis of myocardial function is important for the diagnosis of heart diseases, the planning of therapy [LC99] and the understanding of the effects of cardiac drugs on regional function [Rei99]. This subsection summarizes research on the relationship between cardiac function, wall motion abnormalities and myocardial strain and stress. The results suggest that myocardial stress and strain can be used as an indicator of heart failure.

Myocardial stress and strain are influenced by a variety of determinants including 3D shape (wall thickness, curvature), tissue structure (muscle fiber architecture), pressure, constraints (e.g., due to the pericardium, valves, chordae tendineae) and material properties [Bro00a].

Effects of myocardial stress and strain include vulnerability to injury (ischemia, arrhythmia, cell dropout, aneurysm rupture), remodelling (hypertrophy, fibrosis, scar formation), progression of disease (transition from hypertrophy to failure), ventricular dilation, infarct expansion, response to reperfusion and aneurysm formation [Bro00a]. Conversely many cardiac disorders result in regionally altered myocardial mechanics, i.e., they effect the myocardial strain and stress.

Traditionally an abnormal contractile function of the ventricles has been determined by measuring the wall thickening using cine MRI images (a series of rapidly recorded multiple MRI images) [SSM+87], echocardiography [MSC+86, ASF+94, CAA+97, ABK+00] and SPECT [ASF+94]. Reported wall thickening rates during systole for a healthy heart vary from 40% [BBK+97] to 80% [HBd+97]. Detectable abnormalities include reduced wall thickening after myocardial infarction [HBd+97], regional wall thinning of an infarcted area and compensatory wall thickening and hypertrophy, and left ventricular enlargement (remodelling) [ASF+94, pp.648].

Changes of the wall thickness, however, are only one indicator of impending
heart failure. Willenheimer et al. report that impaired longitudinal motion is also a predictor of heart failure [WCEI97]. Alexander et al. mention that length changes during the heart cycle are bigger for a failing ventricle [ASF+94, p.703]. De Simone et al. report that a subnormal left ventricular midwall fiber shortening indicates a high risk of cardiovascular events in patients with a high blood pressure (hypertensive) even if other indicators still show healthy values [dDK+96, SDd+97]. Guttman et al. observe that if a small section of the heart muscle is experiencing reduced blood flow (ischemia) it might stop participating in the contractile function even though the rest of the heart muscle works normally. Such abnormalities in the deformation behaviour occur even before first symptoms of a heart attack materialize [GZM97]. Additional studies have shown that a site of myocardial ischemia is more likely to experience a myocardial infarction [NHB+97, VPC+96].

A full description of the deformation behaviour of the myocardium is therefore desirable. Such a description is given by the strain tensor field.

The concept of myocardial strain was originally introduced by Mirsky and Parmley [MP73]. As described in section 2.3.1 strain is defined as the pure deformation (without translation and rotation). Scalar strain values can be derived from the strain tensor to quantify the length change of an infinitesimal material volume in a given direction (e.g., the circumferential or radial direction of the ventricle). Negative strain values are interpreted as a local shortening of the myocardium and positive strain values as a local elongation.

The understanding of the stress field in the myocardium can give further insight into the performance of the heart. For example, Heimdal et al. report that the stress-strain relationship more selectively describes the overall tissue characteristics than the pressure-volume relationship [HSTS98]. Taber mentions that an improved understanding of the forces may be useful for research in tissue engineering, infarct healing, and other related areas [Tab01]. Finally McCulloch and Mazhari [MM01] suggest several possible roles of strain and stress measurement in clinical diagnosis.

The influence of mechanical forces on cardiac activity is subject to ongoing research. It has been shown that mechanical forces are related to cardiac electrical activity [KHN99] and that they play a role in the development and remodelling of the cardiovascular system [Tab01]. Humphrey and Yin report that mechanical stress in the heart can affect coronary blood flow, myocardial oxygen consumption, the development of hypertrophy, the stability of cardiac aneurysms and overall cardiac performance [Yet89, chapter 3]. Several authors have examined the effect of stress on a cellular level. As a result it was found that the hypertrophic response to overload can lead to cell elongation (by causing sarcomeres to be added in series) and hence to an increase in wall tension [Ome98, ASF+94]. Fiber length increases linearly with wall force [ASF+94, p.691,p.694]. Furthermore it has been demonstrated that shear stress leads to changes in the paracellular transport which affects the supply of nutrients to the tissue [DCG+01].

Due to these complex relationships myocardial forces play a role in the development and progress of various cardiac diseases. For example, greater systolic wall stress and oxygen consumption after acute myocardial infarction are contributors to
Although it is possible to directly measure regional myocardial strain there is no method for directly measuring myocardial stress. Given the complex geometry, non-linear material properties, large deformations and complex tissue microstructure of the heart, regional stress can only be estimated by solving the equations of finite elasticity using the finite element method [MM01]. The left ventricular finite element model introduced in subsection 3.2.8 can be directly used in the finite element method to solve for stress, motion and material properties. This computational analysis is however outside the scope of this thesis.

Combining the visualization of myocardial strain and stress can provide insight into the structural basis of regional dysfunction under conditions such as acute myocardial infarction and ischemia-reperfusion [MM01]. The next subsection gives an overview of popular techniques for strain measurement and indicates some issues relevant to the computation of myocardial stresses.

### 3.2.5 Measurement of Myocardial Strain and Stress

The deformation of the heart muscle, and therefore the myocardial strain, can be measured by placing markers on or into the heart wall and then tracking them during the heart cycle. Initial methods were invasive and included using optical markers on open chest animals and using radiopaque markers on closed chest animals ([MZ91, HSTS98]) and on humans [IDSA75].

In order to measure the myocardial strain in humans only a non-invasive technique is acceptable. This was first accomplished by using tagged MRI [ZPR+88, AD89] in which a non-invasive pattern of tissue-spin polarization changes is placed and tracked in the myocardium. By using multiple image planes in 3D the displacement vector and strain tensor at each point of the myocardium can be determined [McV96, MO01].

A recently proposed alternative method does not trace individual tag lines but computes 2D deformation information from the Fourier transform of the tagging image [OKMP99]. Aletras et al. do not use tag lines but measure displacement directly by using phase contrast MRI [ABW99]. The method is not susceptible to through-slice motion problems but suffers from a low temporal resolution.

Other image modalities can also be employed: echocardiography has been utilized for strain measurement either by using strain rate imaging (SRI) [HSTS98, EGB+99, SBA+01] or by estimating a motion field using a linear elastic model [PSDD99, PSDD01]. Echocardiography achieves an improved temporal resolution of about 70 frames per second compared with 20 frames per second for the fastest MRI methods. However, the method has a lower signal to noise ratio, is less accurate, only a few strain components can be measured, and the measurement is dependent on the acoustic window [You00]. In addition to MRI and echocardiography left ventricular motion has been assessed using SPECT [FCF+99] and PET [BBK+97]. Both of these methods require the injection of radionuclides, suffer from a restricted ability to assess deformation, and have a limited temporal and spatial resolution if compared
with echocardiography [PSDD01]. In this thesis myocardial strain fields were created using tagged MRI which is explained in the remainder of this subsection.

The first step necessary for measuring myocardial strain is the creation of a non-invasive tagging pattern in the myocardium by generating a rectangular grid of tag planes which are orthogonal to the image plane. The tag planes appear then on the image plane as a rectangular mesh of dark lines [MZ91, FMS+94]. Alternatively a set of coplanar or radial tag planes can be chosen which appear as parallel or radial lines in the image plane. When the heart deforms the tag lines deform with it and reflect accurately the motion of the myocardium [MRM94, YAD+93].

Figure 3.7 shows tagged MRI images of a mid-ventricular short axis slice for a complete heart cycle. The first image shows the heart at end-diastole and the 10th image shows the heart at end-systole. The annulus shape in the centre of each image is the left ventricular wall and the arc extending from its left-hand side is the right ventricular wall (see figure 3.1). Starting from the 8 o’clock position moving in clockwise direction the regions of the left ventricular wall are the interventricular septum, the anterior wall, the lateral wall, and the inferior wall. Note that MR images show all materials with a high water content such as soft tissue and blood. The tag lines intersecting the ventricular cavities are therefore visible in the first image but disappear rapidly due to turbulent blood flow. Figure 3.8 shows the corresponding images for a mid-ventricular long axis slice for each time step.

Guttman et al. report that tagged MRI provides a good spatial resolution and that displacements as small as 0.1mm can be measured [GPM94]. The main limitation of strain measurement by MRI is a low temporal sampling rate and a long examination time [HSTS98]. Since the temporal resolution of MR images is low the images are obtained over multiple heart beats. The correct temporal sequence is achieved by gating (timing) the MRI machine with the patient’s electrocardiogram after compensating it for magnetohydrodynamic effects [Box99, SB99]. Because of misregistration caused by breathing most modern studies use breath-hold cine MRI which captures a complete heart cycle during a single breath-hold while maintaining high resolution across the tag lines [MA92]. A modification to prevent the fading of tag lines during the heart cycle (CSPAMM) is described in [FMS+94]. MRI studies over longer time-spans can be achieved using cardiac-respiratory gating [YAH+00]. Real time MR imaging, however, is already becoming a reality [POCD99] and might soon extend to tagged MRI.

Reconstructing the 3D motion of the heart from tagged MRI images is difficult since the heart moves through the fixed image planes (cardiac through-plane motion). Different regions of tissue are therefore sampled at different times. Moore et al. report that this effect must be corrected even if only performing a 2D analysis of wall deformation [MOMZ92]. In order to compute the deformation most studies use three orthogonal arrays of image planes each with a parallel pattern of MRI tags. Alternatively a grid pattern of MRI tags can be employed as shown in figures 3.7 and 3.8. In this case one set of short axis images and one set of long axis images are sufficient to reconstruct the strain field in the myocardium.

The process of reconstructing the myocardial strain field is divided into three
Figure 3.7. Tagged MRI images of a mid-ventricular short axis slice for a complete heart cycle (ordered from left to right and top to bottom). The first image shows the heart at end-diastole and the 10th image shows the heart at end-systole. The annulus shape in the centre of each image represents the left ventricular wall [The images were produced with our toolkit from tagged MRI data provided by Alistair A. Young].
Figure 3.8. Tagged MRI images of a mid-ventricular long axis slice for a complete heart cycle. The first image shows the heart at end-diastole and the 10th image shows the heart at end-systole. The left ventricular cavity is the semi-ellipsoidal gray shape on the right hand side of each image [The images were produced with our toolkit from tagged MRI data provided by Alistair A. Young].
The first step uses image processing algorithms to detect the tag lines. Guttman et al. [GPM94, GZM97] determine the myocardial contours in a user-specified circular region of interest by removing tags using morphological closing and by employing a graph-search technique. A target tag line pattern precomputed by a simulation is then used to find the correct tag lines inside the myocardial contours. The tag lines are indexed and inconsistencies over different slices or time frames are corrected by manual editing in 2D or 3D [GZM97].

In the second step smooth curves are fitted to the tag lines. A popular tool for this are snakes (active contour models [KWT87]) which minimize both curve-bending energy and a potential which is lowest at “wall-like points” [MOMZ92, SPH+96]. Manual editing of the results is often necessary (usually due to the presence of papillary muscles on the inner wall boundaries). Since errors in slices are best recognized in comparison with neighbouring images Solaiyappan et al. allow an editing of the images in 3D [SPH+96]. Amini et al. modify this approach and replace the individual B-Splines with a 2D coupled B-Spline grid [ACC+98]. Moulton et al. use a semiautomated method to approximate tag lines by third-order B-Spline curves [MCD+96].

Finally the individual tag lines are used to reconstruct the 3D deformation field. Early methods reconstructed 3D motion by using identifiable points within the images such as intersections between tag lines [YAD+93], intersections between tag lines and myocardial contours [MOMZ92], or points along striped tag lines. Such schemes neglect most of the information in the tagged images and suffer from poor spatial resolution [OMH+95]. O’Dell et al. [OMH+95] compute the displacement field at \( t = \tau \) by associating each tag line at a time \( t = \tau \) with a tag line at \( t = 0 \). For each of the three orthogonal image planes the authors can then compute one-dimensional displacement values as illustrated in figure 3.9. The resulting three sets of independent one-dimensional displacement data are then least square fitted to an analytical series in prolate-spheroidal coordinates. The resulting analytic function describes the displacement field at \( t = \tau \). Ozturk and McVeigh [OM00, MO01] also measure the 1D displacements of individual tag lines but employ an image plane based cartesian coordinate system to describe the heart motion using a 4D tensor product of B-Splines.

Once the displacement field \( u \) of the myocardium is computed different methods (depending on the representation of \( u \)) can be employed to derive the strain field. Most authors define the strain by the Lagrangian finite strain tensor \( E \) [DM97, MCD+96, OMH+95] which was explained in subsection 2.3.1 and can be computed by

\[
E = \frac{1}{2} (F^T F - I) = \frac{1}{2} \left[ \nabla u^T + \nabla u + \nabla u^T \nabla u \right]
\]

where \( F = I + \nabla u \) is the deformation gradient tensor and \( \nabla u \) is the displacement gradient tensor for the motion from the undeformed to the deformed state (equation 2.10).

The resulting models can be validated using thick-walled cylinders under inflation and torsion [YAD+93], thin plates under tension [HYMW93], and finite element simulations of the left ventricle [MCD+96]. A simulation toolkit for two-dimensional
Figure 3.9. For each image plane and each stack of tag planes one-dimensional displacement values $\Delta x$ can be measured along each tag line.

MRI has been presented by Crum et al. [CBR+97]. Various measures can be derived from the strain tensor. The principal strains give the direction and magnitude of the maximum and minimum strains (see subsection 2.3.1). Alternatively strains in any direction, such as the radial (transmural) direction can be computed by multiplying the strain tensor with the normalised direction vector. Azhari et al. [AWR+95] define the area strain $\Omega$ as the amount by which an infinitesimal surface element has shrunk, i.e.,

$$\Omega = 1 - \sqrt{2E_1 + 1} \sqrt{2E_2 + 1}$$

where $E_1$ and $E_2$ are the principal Lagrangian strains of the examined surface. The authors perform a comparative study of myocardial strains in ischemic canine hearts and report that the best indicators of ischemic tissue are the endocardial area strain, the endocardial principal strain and the transmural wall thickening with the first measure being the best. All of these measures were shown to be superior to assessing wall motion by echocardiography [AWR+95, vR99].

Mechanical stresses cannot be measured directly in the intact heart since measuring the force at a material point by using a transducer would affect the stress distribution in the myocardium. Instead stress must be inferred from biomechanical analysis [MSTM01, RKN00] which is based on descriptions of the material properties [Yet89, HNS+93]. Several research groups have recently published results obtained with finite element analysis [GCM95, CHM01, MM01].

3.2.6 Previous Studies of Myocardial Strain

Myocardial strain has been studied both for healthy subjects and for a variety of cardiac diseases. This subsection summarizes results obtained for healthy hearts.
Myocardial Strain in the Healthy Left Ventricle

The strain in the myocardium is highly inhomogeneous and anisotropic [HSTS98]. The strain behaviour of properly functional myocardium is a thickening (stretch) in the radial direction and a shortening (compression) in the circumferential and longitudinal directions [GZM97]. The eigenvectors of the strain tensor give the direction and amount of the maximum shortening and the maximum stretch and are normally oriented approximately in the circumferential and radial directions, respectively [GZM97, YICA94]. The radial and circumferential strains increase in magnitude from apex to base, from endocardium to the epicardium, and from the septum to the free wall [DM97].

Reichek [Rei99] notes that the transmural gradient in the circumferential strain is surprising since the maximum circumferential strain would be expected to be in the fiber direction. However this is not the case: the fiber orientation is closest to the circumferential direction at midwall whereas close to the epi- and endocardial surface it is directed more towards the long axis. The fiber strain is relatively uniform across the wall and the author suggests that the transmural gradient in the circumferential strain is due to cross-fiber strain.

Young et al. [YICA94] examine the 2D strain for SA images and report that at the base the maximum principal strain at the lateral wall is $0.29 \pm 0.05$ and hence bigger than at the septum and the anterior wall (0.21 ± 0.08 and 0.21 ± 0.06). Extension is greater at the base (0.24 ± 0.07) than at the apex (0.15 ± 0.07). The minimum principal strain increases in magnitude from the base ($-0.19 \pm 0.03$) towards the apex ($-0.22 \pm 0.02$) with little circumferential variation, except at midventricle where the maximum contraction in the anterior region is $-0.21 \pm 0.03$, slightly higher than in the posterior region with $-0.19 \pm 0.02$. The contractions in longitudinal and circumferential directions are similar and show little variation except that shortening in the septum increases towards the apex.

O’Dell et al. [OMH+95] who report that for healthy volunteers the ranges of circumferential, longitudinal, and midwall radial strain are (-0.25,0.06), (-0.23,-0.08), and (0.18,0.52). Lugo et al. [LOMP+94] report that for the healthy heart the circumferential and longitudinal strains are approximately 30-40% greater in the anterior wall than in the septum while the radial strain is nearly homogeneous.

Few investigators give data about the shear strain in the myocardium. The human heart model of Moulton et al. [MCD+96] gives shear strains of between -0.1 and 0.1 which is similar to measurements for the canine heart done by Omens et al. [OMM91]. Young et al. observe that the shear strain $E_{rc}$ changes sign from apex to base [YICA94].

Several authors prefer to measure the percentage segment shortening in the circumferential and longitudinal directions rather than the strain. The reason for this is its conceptual familiarity to cardiologists and its analogy to data derived by sonomicrometry and other methods [Rei99]. The negative eigenvalues $\lambda$ of the Lagrangian strain are related to the percentage segmental shortening ($%S$) in the corresponding
The Human Heart

principal direction by \( \%S = (1 - \sqrt{2\lambda + 1}) \times 100\% \) [YICA94]. The relationship between the eigenvalues \( \lambda \) and the corresponding principal stretch \( L = \frac{l_{deformed}}{l_{undeformed}} \) is given by \( \lambda = \frac{1}{2}(L^2 - 1) \).

Young et al. [YKF+94] report that circumferential shortening typically ranges from 17 to 21\% consistent with shortening of the circumferential muscle fibers. An older study by Clark et al. [CRB+91] which we believe is less reliable reports the circumferential shortening at the epicardium to be 22 ± 5, in the midwall 30 ± 6\% and at the endocardium 44 ± 6\%. Shortening at the midwall and endocardium is at the base 5-10 percentage points smaller than at the apex. The shortening in the anterior region is lowest and in the septal region highest.

Myocardial Strain in a Heart with Dilated Cardiomyopathy

Dilated cardiomyopathy is characterized by cardiac enlargement, increased cardiac volume, reduced ejection fraction, and congestive failure. Large LV dilation and wall thinning can be recognized on MRI images, with RV and atrial dilation sometimes also being present. MRI tagging shows reduced cross-fiber shortening at the endocardium due to an underlying myocardial fibrosis and increased end-systolic wall stress [SB99]. Reduced fiber and cross-fiber strain with a preserved transmural gradient has also been reported [Rei99].

Young et al. [YDP+00] present a detailed examination of the myocardial strain in left ventricles with non-ischemic dilated cardiomyopathy. The authors report a consistent strong regional heterogeneity with systolic lengthening in the septum of \(-5 ± 7\%\) in the circumferential direction and \(-2 ± 5\%\) in the longitudinal direction. In contrast the lateral wall showed relatively normal systolic shortening of \(12 ± 6\%\) in the circumferential direction and \(6 ± 5\%\) in the longitudinal direction.

3.2.7 The Visualization of Myocardial Strain and other Functional Values

In recent years an increased effort has been made to visualize cardiac deformation and strain and a summary is given below. Reichek notes that further progress in the development of effective software analysis packages for tagged MRI is badly needed to facilitate a faster and more flexible data display and interpretation and to circulate the approach more widely [Rei99].

Solaiyappan et al. visualize tagged MRI images in real time directly using 3D texture mapping hardware [SPH+96]. Using a real-time approach has the advantage that an MRI scan can be extended if more information is required reducing the need of multiple examinations. The authors motivate their visualization by demonstrating that a region of non-moving MRI tags is perceived easier than a non-contracting section of the heart wall. High performance is achieved by dividing the MRI data into blocks and by limiting the number of MRI slices used in the texture mapping procedure according to the amount of user interaction. Perception of the data is improved by clipping the MRI volume to an ellipsoidal shape intersecting
the myocardium. For non-real-time applications additional information is displayed by embedding polygonal surfaces (such as the ventricular surfaces) into the volume. Editing of and interaction with the MRI data is done via a virtual workbench.

McVeigh et al. use direct volume rendering to display the tagged MRI raw data \[\text{MGP}^+94\]. The authors also visualize the strains in the radial, circumferential and longitudinal direction within a left ventricular model by colour mapping them onto cylindrical shells.

Guttman et al. visualize the strain field in a model of the left ventricle obtained from MRI images \[\text{GZM97}\]. The authors represent the strain tensor with respect to a cylindrical coordinate system and display its circumferential, longitudinal or radial components over a chosen area using Gouraud shaded polygonal surfaces. A region of interest is selected using range values, clipping planes and by specifying shells inside the heart’s wall. Principal strains are visualized using colour coded line icons. All of the above visualizations can be animated. The authors emphasize the importance of combining the derived functional parameters with the raw image data in a 3D display to give them anatomical context. Displayed MRI slices are animated and can be changed without pausing the movie. Using 3D texture mapping hardware the authors can move through the beating heart along the long axis by bilinear interpolating between short axis images. Animations for different models are either played simultaneously (same time scale) or synchronously (adapting to different heart rates). A single raw image can be enhanced by displaying simultaneously vector plots and translucent colour coded scalar data.

Chernoff and Higgins prefer strain maps over the 3D visualization of the strain field. Strain maps are collections of 2D plots of scalar strain values (such as principal strains or normal strains) over time for each LV region. The plots are ordered in a 2D array with each array element corresponding to different myocardial regions in the longitudinal and circumferential direction. Age and gender matched normal curves at ±2 standard deviation are displayed simultaneously with the patient’s data. The location of an abnormality is then given by the plot where the patient’s data lies outside the standard deviation curves \[\text{SB99, chapter 21}\].

The visualization methods reviewed so far use only the principal strains or the strains in the cardiac specific axes (i.e., radial, circumferential and longitudinal direction). Reichek suggests as an innovative approach of data analysis to relate the endocardial and epicardial surface strain to the fiber orientation and to compute which proportion is due to fiber shortening (fiber strain) and remotely generated forces (cross-fiber strain). Sinusas et al. \[\text{SPC}^+01\] also compute and visualize the strain in the fiber specific axes. Recent research towards measuring the myocardial fiber orientation in vivo is summarized at the end of subsection 3.2.1.

As mentioned in the previous subsection currently the reconstruction of the myocardial strain field is done by a semi-automated postprocess. Recent advances in quantitative automated MR image analysis \[\text{vR99}\] and cardiac MRI, such as real-time imaging and interactive plane steering \[\text{Rei99}\], give hope that in future the myocardial strain field can be interactively explored \[\text{WSC97, Rei99}\].

Various other measures and functions have been proposed to visualize myocardial
3.2 The Human Heart

deformation. Young et al. [YICA94] visualize the 2D trajectories of myocardial sample points and van der Geest et al. analyse the strain rate measured by VEC-MRI [vR99]. Clarysse et al. [CFM97] report that the heart function can also be estimated from shape changes of the left-ventricular endocardial surface. The authors use two curvature measures based on the Gaussian and \( K \) and mean \( H \) curvature (see appendix F). The shape index \( s \) is defined as

\[
s = \frac{2}{\pi} \tan^{-1} \frac{k_2 + k_1}{k_2 - k_1} = \frac{2}{\pi} \tan^{-1} \frac{-H}{(H^2 - K^2)^{\frac{1}{2}}} , \quad k_1 \geq k_2
\]

and the curvedness \( c \) is defined as

\[
c = \left( \frac{k_2^2 + k_1^2}{2} \right)^{\frac{1}{2}} = (2H^2 - K)^{\frac{1}{2}}
\]

where \( \kappa_1 \) and \( \kappa_2 \) are the principal curvatures. The shape index \( s \) gives a continuous distribution of shape types ranging from \(-1\) (cup) through \(-0.5\) (valley), \(0\) (symmetrical saddle), \(0.5\) (ridge) to \(1\) (peak). The shape index of a planar point is undefined. The expression \( c^2 \) represents the potential energy of an ideal flexible thin plate and hence gives the local strain energy of the deformation. The authors analyze the heart function by plotting a temporal spectrum based on these two curvature measures. The approach seems to be most useful for temporal data without explicit point-to-point correspondence such as obtained by echocardiography or CT.

Other important parameters which can be extracted from MRI (and other) imaging data are the ejection fraction (percentage of blood in a ventricle at maximum expansion which is ejected during contraction) and the cardiac output (the volume of blood ejected from the left ventricle in one minute). While these measures are recognized as important indices of global ventricular performance [SB99] it has been shown that such global measures are poor predictors in left ventricular function after an acute myocardial infarction [LRMQ01]. Patient therapy depends on knowledge of systolic performance and MRI is a more accurate measure of ventricular performance than echocardiography or ventriculography [SB99]. Another entity frequently visualized by 2D graphs is the wall thickness along short axis images of the left ventricle [vR99].

### 3.2.8 FE Modelling of the Heart

Finite element models of the heart can be obtained either directly from biomedical imaging data or from mathematical simulations. In this research we are predominantly interested in FE models of the human left ventricle and its associated strain field which is obtained from tagged MRI images.

**FE Model of the Left Ventricle**

A model for reconstructing the 3D motion and strain of the left ventricle from MR images has been developed by Young et al. based on a finite element model of the left
ventricle [YA92, YKDA95]. The authors compute the model geometry by tracking myocardial contours on tagged MRI slices and by fitting a surface through them using a prolate spheroidal coordinate system (see appendix F) aligned to the central axis of the left ventricle. The process is illustrated in figure 3.10.

A FE model is created by placing nodes at equal angular intervals in the circumferential and longitudinal direction and by fitting the radial coordinate to the inner and the outer surface. The model is then converted into a rectangular Cartesian coordinate system with the long axis of the ventricle oriented along the x-axis and the y-axis directed towards the centre of the right ventricle. The resulting model consists of 16 finite elements with its geometry being interpolated in the radial direction using linear Lagrange basis functions and in the circumferential and longitudinal directions using cubic Hermite basis functions. In addition to the FE geometry two Bezier surfaces consisting of 16 bicubic patches are provided. The surfaces represent the epicardial and the endocardial surface and are used to compute the scaling factors for the derivatives of the cubic Hermite basis functions (see subsection 2.4.2).

![Figure 3.10. Tag lines and ventricular contours before (a) and after (b) myocardial contraction and the fitted epicardial and endocardial surfaces (c) [With kind permission from Dr. Alistair A. Young ©December 2002].](image)

The authors generate model geometries for 9 time steps equally spaced over half a heart cycle from end-diastole (maximum expansion) to end-systole (maximum contraction). End-diastole is determined by the rising R wave of the ECG, whereas end-systole is defined as the instant of least cavity area in the midventricle [YICA94]. Determining the correct moment of end-systole is difficult because there is a period of isovolumic relaxation lasting 50-100 milliseconds in which both aortic and mitral valves are shut and the volume is constant. The frame at the end of ejection is therefore determined by looking at a cine image sequence [You02].

The tagged MRI images used to compute strain information are produced by creating multiple parallel tagging planes of magnetic saturation orthogonal to the imaging plane in a short time interval (~10 msec) after detection of the R wave. The intersection of these tagging planes with the image plane gives rise to dark stripes ~1 mm in width and spaced ~6 mm apart. The image stripes deform with the
3.2 The Human Heart

underlying tissue and fade according to the longitudinal relaxation time constant T1 (∼800 msec for myocardium) [You02]. The authors track tag lines using a 2D weave of active contours (snakes) and locate tag points within the corresponding 3-D model for each time step (figure 3.10 (a) and (b)). Using a special objective function to fit the 1-D displacements of tag lines back to the undeformed state and minimizing this function makes it possible to reconstruct the 3-D displacement of each material point. The authors fit the original undeformed model to this data to reconstruct the deformation of this model. The strain tensor is obtained from the deformation gradient tensor using equation 3.1. The computation of the strain field was validated by the authors of this model using a gel phantom [YKDA95, KYCA95].

Since the strain field is computed from the deformation between end-diastole and end-systole the case study in chapter 6 uses only the models at these two moments. Images of the model at maximum expansion and maximum contraction are shown in figure 3.11 and 3.12, respectively.

![Figure 3.11. The finite element model of the left ventricle at end-diastole.](image1)

![Figure 3.12. The finite element model of the left ventricle at end-systole.](image2)

The strain field is represented by 10x10x6 sample points per element with 10 sample points each in the circumferential and longitudinal directions and 6 sample points in the radial direction. No strain values are defined along the longitudinal axis where the four apical finite elements meet since the faces of the adjoining elements collapse down to a line on this axis. As a result the elements have a singularity along this axis (the derivative in circumferential direction is undefined) and the displacement gradient is undefined.

In order to get a continuous visualization we generate values for the strain tensor field at any point along the longitudinal axis by averaging the strain values of the point’s neighbours in the longitudinal direction. A linear interpolation is used since
no derivatives of the strain field are known and since the values close to the apex
suffer already from a large error due to the singularity in the model.

Note that at the apex the longitudinal material direction has got the largest
angle with the longitudinal axis whereas the radial direction at that point is parallel
to this axis. A continuous strain field is obtained by trilinearly interpolating the
sample values over each element. Each strain value is defined with respect to the
material coordinate system, i.e., the normal components of a tensor represent the
strains in circumferential, longitudinal and radial direction, respectively.

More information and alternative methods for the computation of myocardial
strains are presented in subsection 3.2.5.

Numerical Model of the Canine Heart

A three-dimensional finite element model of the mechanical and electrical behaviour
of a dog heart has been developed by the Department of Engineering Science and
the Physiology Department of the University of Auckland in collaboration with the
University at San Diego, U.S. and the McGill University, Canada [HNS+93].

The model is based on the theory of large deformation elasticity and is solved
using Galerkin and collocation techniques. Electrical activation is described by
the FitzHugh-Nagumo equations and the mechanical behaviour is governed by an
orthotropic “pole-zero” law and a Wiener cascade model for the passive and active
properties of the myocardium, respectively. Since the behaviour of the myocardium
is highly anisotropic the model incorporates the orientation of the muscle fibers and
the fiber sheet normals. Details are described in [HS88, HNS+93, Bio97]. Relevant
principles of biomechanical modelling are explained in [Bio01, BGL96, Fun90].

The initial version of the model defined the heart geometry using a prolate
spheroidal coordinate system which made it possible to model the ventricular ge-
ometry simply and efficiently using trilinear isoparametric elements [HS88]. An
improved version of this model is described in [GCM95].

Our work uses a more recent version of the model defined in Cartesian coordinates
using isoparametric finite elements with tricubic Hermite interpolated geometry. The
model has 60 elements and 99 nodes and was computed by the Bioengineering Group

Strain values are defined for each node and are trilinearly interpolated over the
elements. In addition the model defines for each node the muscle fiber direction
and the sheet normal direction. The fiber angle is defined with respect to the
circumferential material coordinate direction and is interpolated in the longitudinal
and radial directions using linear basis functions and in the circumferential direction
using cubic Hermite basis functions. The sheet angle is interpolated in the radial
direction using linear basis functions and in the circumferential and longitudinal
directions using cubic Hermite basis functions.

The use of different-order basis functions for dependent and independent vari-
ables is due to different spatial variations and continuity requirements. Note that the
muscle fiber and sheet normal directions are specified with respect to the material
coordinate axes so that their orientation changes consistently with the deformation
of the model. Figure 3.13 shows an image of the model. The white lines represent the finite element mesh, the blue surfaces are the endocardial surfaces of the left and right ventricle and the red line segments indicate the myocardial fiber direction.

![Figure 3.13](image)

**Figure 3.13.** A wireframe representation of the finite element model of the dog heart during contraction. The element faces containing the ventricular cavities are rendered as blue surfaces and the myocardial fiber direction is indicated by red lines. The cavity on the left hand side of the image is the left ventricle.

The literature offers a variety of alternative mathematical heart models and a good overview is given in [Wei97]. Additional information on cardio-vascular modelling is found, for example, in [Bio01, Bio97, CL95, Pow95, Yin85, SW93, SB85].

### 3.3 The Human Brain

This section introduces the anatomy of the human brain and describes diffusion tensor imaging (DTI) which can be used to obtain information about the neuroanatomy in vivo. The subsequent subsections summarize various measures used to express anatomical properties and present a survey of previously employed visualization methods for DTI data. We conclude with a description of the DTI data set used in the case study in chapter 7.
3.3.1 The Anatomy and Physiology of the Brain

The brain is a part of the central nervous system and is responsible for storing, evaluating and reacting to sensory information from the external and internal environment. Information is received from the endings of special sensory nerves in the skin, the deep tissue, the eyes, the ears and in other sensors, and is then transmitted via the spinal cord to the brain (sensory nerves inside the head are directly connected to the corresponding processing area in the brain). The principal functions of the brain are sensory function (information flow from body to brain), integrative function (brain to brain), which includes the memory and thinking processes, and motor function (brain to body) [Guy87]. This subsection presents an overview of the brain anatomy and physiology using references from [Guy87, EW91, JB, HWGR98, Abo].

**Terminology**

A special terminology is used to describe directional information in relation to the brain inside the skull. The terms, illustrated by figure 3.14 are: anterior for towards the face, posterior for towards the back of the head, lateral for towards the sides, superior for towards the top of the skull and inferior for towards the skull’s base.

The centre of the brain, defined with respect to its development, is given by the neuraxis (figure 3.15). Since the brain is bending during development the neuraxis consist of two straight line segments which define a plane called the median plane. Sections parallel to this plane are called sagittal. Sections orthogonal to the median plane and parallel to the horizontal segment of the neuraxis are called horizontal and sections orthogonal to both sagittal and horizontal sections are called coronal.

**Nervous Tissue**

In order to identify anatomical structures a basic knowledge of the composition of brain tissue is necessary.

Brain tissue consists of nerve cells (neurons) and supporting cells (neuroglia). Each neuron has a cell body and several processes which are classified as dendrites or axons, the latter of which are usually long, single and clearly separated from the cell body. The nerve cells form two types of brain tissue: gray matter and white matter. Gray matter consists of neural cell bodies, the surrounding neuroglia, and an intermingling of axons and dendrites and their connecting contacts (synapses) where information is transmitted. White matter consists only of nerve fibers, each of which consists of an axon and its supporting cells. Bundles of nerve fibers with common origin and destination are called nerve fiber tracts or just fiber tracts.

Information is transmitted using action potentials traveling down the axon by jumping over gaps (nodes of Ranvier) in the myelin sheaths, which surround and insulate the axon. The process is called saltatory conduction and is considerably faster (35-60 m/sec) than if using a continuous depolarisation [SE00].
3.3 The Human Brain

Anatomy and Physiology

The human brain is located in the cranial cavity and consists of the cerebrum, the cerebellum, and the brain stem (figure 3.14 and 3.15). The cerebrum makes up the largest portion of the brain and has the shape of two symmetric “squashed” hemispheres. Each hemisphere consists of four lobes indicated in figure 3.14 with black curves. The image shows in clockwise direction starting from the left the frontal, parietal, occipital, and temporal lobe.

The outer 2 – 4mm wide layer of the cerebrum is formed by gray matter and constitutes the cerebral cortex. It is responsible for storing memories and in combination with other structures for thinking, feelings and fine motoric movements. The only other areas of gray matter in the cerebrum are the basal ganglia, which are located deep within the cerebral hemisphere (see figure 3.16) and are responsible for cognition, movement coordination and voluntary movement. The basal ganglia consist of the caudate nucleus, the putamen and the globus pallidus.

The cerebellum is located inferior to the occipital lobe of the cerebrum and is connected to it by the midbrain (mesencephalon) which is the top most part of the brain stem. The cerebellum is responsible for the coordination of muscle contractions during complex movements and other mainly motoric functions. The
midbrain controls responses to sight, eye movement, pupil dilation, body movements and hearing. Superior to the midbrain is the thalamus which is a large dual lobed mass of gray matter and is responsible for motor control and the reception of auditory and visual sensory signals. Below the thalamus is the hypothalamus which controls autonomic and endocrine functions and regulates food and water intake.

Gray matter areas in the cerebrum, the cerebellum and the brain stem are connected by white matter which in some regions is bundled together to form fiber tracts. The major principal pathways consist of millions of nerve fibers and can be differentiated into commissural fibers, association fibers, and projection fibers. The major fiber tracts are explained in the following paragraphs and are shown in figure 3.15 and 3.16.

Commissural fibers cross over or join the two halves of the brain communicating between identical cortical areas in either hemisphere. The largest of these is the corpus callosum which runs over the top of the lateral ventricles and has a genu and a splenium.

Association fibers enable the communication between different areas of the cortex in the same hemisphere, and allow the integration of information from different
3.3 The Human Brain

Figure 3.16. Horizontal section through the brain at the position of the neuraxis (the image was produced using a $T_1$ weighted MRI data set and a volume visualization program obtained from [RSEB+00]).

1. Fornix
2. White matter (cerebral cortex)
3. Grey matter (cerebral cortex)
4. Corpus callosum ( genu)
5. Corpus callosum (splenium)
6. Lateral ventricle (anterior horn)
7. Lateral ventricle (posterior horn)
8. Basal ganglia (caudate nucleus)
9. Basal ganglia (putamen and globus pallidus)
10. Internal capsule
11. Thalamus
12. Optic radiation

Parts of the cortex. Examples are the superior longitudinal fasciculus which interconnects Broca’s area of motor speech with Wernicke’s area of language perception, the inferior occipito-frontal fasciculus which connects the occipital and temporal lobes with the frontal lobes, and the fornix which connects the hypothalamus to the cerebrum.

Projection fibers convey sensory information from the body to the cortex (sensory nerve fibers) and motor information from the cortex down into the brain stem and spinal cord (motor nerve fibers). The major projection system is the internal capsule which is radially arranged where it leaves (or enters) the cortical mantle. The internal capsule continues as corona radiata in the superior direction and forms the cerebral peduncles in the inferior direction where it is joined by the external capsule.

The brain stem connects the forebrain with the spinal cord and comprises the medulla oblongata, the pons and the midbrain as shown in figure 3.15. Apart from its connective function the brain stem also contains gray matter areas which control physiological variables such as arterial pressure, respiration and equilibrium.

The brain is surrounded by and suspended in cerebral spinal fluid (CSF) which
Biomedical Structures, Models and Data Sets

protects the brain by absorbing shocks. CSF is also found in a series of interconnected cavities (ventricles) shown in figure 3.17. Blood is supplied to the brain through a network of capillaries coated by astrocytes which regulate what enters the brain (blood-brain barrier).

Figure 3.17. Posterior view of a cast of the ventricles in the human brain. The structures numbered 2-5 form the lateral ventricles (©1991 Wolfe Publishing Ltd. [EW91]).

1 Cerebral aqueduct
2 Anterior horn
3 Body
4 Posterior horn
5 Inferior horn
6 Third ventricle
7 Fourth ventricle

3.3.2 Diffusion Tensor Imaging

Diffusion tensor imaging (DTI), also known as diffusion-weighted MRI imaging (DWI), is used to measure the intrinsic properties of water diffusion in the brain by an orientation invariant quantity, the diffusion tensor \( D \) [BMB94, Bas95].

The eigenvalues and eigenvectors of the symmetric second-order tensor \( D \) define the principal axis of a diffusion ellipsoid which expresses the spatial distribution of water molecules originating at a point location after an infinitesimal time period.

DTI almost completely suppresses water in the blood vessels [Bas00] and can be used to measure the diffusion of cerebral spinal fluid (CSF) and fluid inside of nerve cells. The results of the measurement are the six components of the symmetric

\[ D = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{pmatrix} \]

\(^{1}\)Since a tensor is independent of the coordinate system used the measured diffusion tensor does not depend on the device coordinates of the MRI machine as would be the case if, for example, measuring the water diffusion in \( x \)-direction.
diffusion tensor $D$ and the $T_2$ weighted signal intensity in the absence of diffusion sensitization. Images of water diffusion can provide pathophysiological information complementary to $T_1$ and $T_2$ weighted MRI images. The technique is sensitive to movements of the order of a few microns and is described in more detail in [BP96a, PJB+96, Hedb, AKM+99].

In the brain DTI can be used to differentiate three types of structures. Fluid filled compartments are characterized by a very high isotropic diffusion, i.e., the diffusion is similar in all directions. In contrast nerve fibers restrict the diffusion to one direction only due to the presence of cell membranes and myelin sheaths surrounding the axons. Fiber tracts, consisting of millions of parallel nerve fibers, are therefore identified as areas of a high anisotropic diffusion. The orientation of such fiber tracts is determined from the principal directions of the diffusion tensor (see section 2.1). Finally gray matter is characterized by a low and nearly isotropic diffusion since the water diffusion is restricted in all directions due to cell membranes of intermingled cell bodies and their surrounding neuraglia. Consequently DTI can be used to gain in vivo information about the anatomy, microstructure and physiology of the brain.

DTI imaging has been successfully applied to diagnose various diseases. For stroke victims it has been shown that diffusion reduces in the demarcated ischemic region within minutes whereas changes in conventional $T_2$ weighted MRI images become apparent only after about three hours [WCL+92, Zag]. Also it has been reported that the fractional anisotropy (see next section) in white matter regions decreases for at least four weeks in correspondence with the theory of its structural degeneration which is not apparent in conventional MRI images [Heda]. The pathologic basis for diffusion changes in the ischemic brain is still subject to controversy [Zag].

For schizophrenic patients it has been found that the fractional anisotropy in the frontal lobes is reduced despite having no significant volume deficit [LHM+99]. Assaf et al. report that DTI images are also sensitive to the pathophysiological state of white matter in brains diagnosed with Multiple Sclerosis [ABB+02]. Zhang et al. note that the white matter regions adjacent to the edema surrounding a metastasis are characterized by heterogeneity in the diffusion anisotropy [ZLB+02a, ZLB+02b]. Barnea-Goraly et al. show that regionally specific alterations of white matter integrity occur in patients with Fragile X Syndrome, a common form of hereditary mental retardation [BGEH+03].

Inder et al. investigate periventricular leukomalacia (PVL), the principal form of brain injury in the premature infant [IHM+00]. The authors report that the diffuse cerebral white matter injury associated with PVL could not be consistently detected early with conventional imaging techniques but shows up as a striking bilateral decrease in water diffusion in cerebral white matter using DTI. DTI has also been used to investigate the development of white matter tracts in adolescents and adults [LN02].

The above examples demonstrate the importance of diffusion tensor imaging for medical diagnosis and research. It is important to note that, since the resolution of DTI is limited, small fibers adjacent to each other and branching fibers cannot be
distinguished. Recent research attempts to improve the standard diffusion tensor model by using high angular resolution diffusion weighted acquisition [OVS+01].

3.3.3 Derived Quantities

The matrix representation of a second-order tensor depends on the coordinate system used (MRI coordinates). In order to describe intrinsic tissue properties variables independent on the patient’s position must be derived. Examples are the eigenvalues and eigenvectors mentioned in the previous subsection and the three tensor invariants (see subsection 2.1.1).

To facilitate the definition of new measures it is convenient to order the three eigenvalues of the diffusion tensor $D$ by size with $\lambda_1$ being the biggest and $\lambda_3$ being the smallest [PJB+96]. The maximum diffusivity is then given by $\lambda_{\text{max}} = \lambda_1$.

The mean diffusivity is defined as the average eigenvalue of the diffusion tensor and is efficiently computed by using the first tensor invariant (equation 2.4)

$$\lambda_{\text{mean}} = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3} = \frac{\text{trace}(D)}{3} = \frac{D_{11} + D_{22} + D_{33}}{3}$$  \hspace{1cm} (3.2)

Images of $\lambda_{\text{mean}}$ show all brain tissue and fluid filled compartments. Note that the computation does not require the computation of the eigenvalues but involves merely averaging the diagonal elements of the tensor matrix.

Another important measure is the anisotropy of the diffusion tensor. Pierpaoli et al. [PB96] define the anisotropy ratio as

$$\lambda_{\text{anisotropyRatio1}} = \frac{\lambda_1}{\lambda_3}$$

which gives the relative magnitude of diffusivities along fiber-tracts and a traverse direction. Alternative measures are

$$\lambda_{\text{anisotropyRatio2}} = \frac{\lambda_1}{(\lambda_2 + \lambda_3)/2}$$

and the fractional anisotropy

$$\lambda_{\text{frac}} = \frac{\sqrt{3}}{\sqrt{2}} \left( \frac{(D - \lambda_{\text{mean}}I) \otimes (D - \lambda_{\text{mean}}I)}{D \otimes D} \right)$$

where the authors define the operator $\otimes$ by

$$D \otimes D = \sum_{i=1}^{3} \sum_{j=1}^{3} D_{ij}^2$$

The fractional isotropy has the property that $0 \leq \lambda_{\text{frac}} \leq 1$ and it is zero if all eigenvalues are equal (perfectly isotropic) and one if two eigenvalues are zero and one not equal to zero (perfectly anisotropic).
3.3 The Human Brain

A further anisotropy measure introduced by [PB96] is the volume ratio

\[ \lambda_{\text{volumeRatio}} = \frac{\lambda_1 \lambda_2 \lambda_3}{\lambda_3^{\text{mean}}} = \frac{27 \det(D)}{[\text{trace}(D)]^3} \]

where \( D \) is the diffusion tensor and \( \det(D) \) and \( \text{trace}(D) \) are the determinant and trace of \( D \), respectively. The volume ratio defines the ratio of the volumes of an ellipsoid whose principal axes are given by the principal diffusivities and a sphere whose radius is the mean diffusivity. The ratio is therefore always in \([0, 1]\).

Alternative measures have been proposed by Westin et al. [WPG+97]. The authors define a linear isotropy \( c_l \), a planar isotropy \( c_p \), and an isotropy \( c_s \) as

\[ c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \] (3.3)

\[ c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3} \] (3.4)

\[ c_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \] (3.5)

The measures fall in the range \([0, 1]\) and sum up to 1 and define therefore a barycentric space of anisotropies. It is also possible to define an anisotropy index as

\[ c_a = 1 - c_s = c_l + c_p = \frac{\lambda_1 + \lambda_2 - 2\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \]

An extensive survey and evaluation of anisotropy indices is given in [SLNI00].

3.3.4 Analysis and Visualization of DTI Data

Originally work in DTI concentrated on the statistical evaluation and the segmentation of DTI data and their correlation with anatomical features. For example, Pierpaoli et al. [PJB+96] use statistical voxel based methods to identify regions of anisotropy, oblate anisotropy, cylindrical anisotropy and asymmetric anisotropy and determine the significance of differences in these measures between various white matter regions. The authors identify isotropic diffusion in the frontal cortex and the caudate nucleus, cylindrical anisotropic diffusion in the pyramidal tract and the corpus callosum and asymmetric anisotropic diffusion in the centrum semiovale, u-fibers, optic radiation and internal capsule. Peled et al. measure the brain anisotropy in various regions of interest and report higher anisotropy values for fiber tracts in the right hemisphere [PGW+98].

Pierpaoli et al. use the mean diffusivity and diffusion anisotropy to segment MRI slices into white matter, gray matter and CSF [PJB+96]. The classification contains ambiguities and we found in the literature no comparison with classification algorithms using traditional MR image modalities such as \( T_1 \)-weighting [MvD+99], \( T_2 \)-weighting and proton density weighting [WWG+99].
Because of noise and the low sample density of DTI it is usually necessary to smooth, regularize and reconstruct the data before visualizing and analysing it. Suitable techniques are described in [AB99, WMK+99, HPP01, XMSD02]. Distortions induced by eddy-currents are characterized and corrected in [JBP98, MPC+01].

Visualization of DTI Data

Diffusion tensors in the medical field have been originally visualized in two dimensions by representing a derived scalar measure over a data slice with a colour map or a gray scale map [PJB+96]. Subsequent research has examined the visualization of directional tensor information over 2D slices via colour mapping [JWH97, Pie97]. Peled et al. visualize a slice of a diffusion tensor data set by indicating the in-plane component of the principal diffusion with a blue line and the out-of-plane components by colours ranging from green through yellow to red [PGW+98].

The idea can be extended to 3D in order to volume render fiber tracts. Several authors have proposed assigning different weightings for red, green and blue colour components according to the X, Y, and Z component of the fiber orientation within a voxel [WLW00, XMSD02]. Pajevic and Pierpaoli investigate the use of color to represent the directional information contained in the diffusion tensor and argue that no scheme exists in which differences in the orientation of anisotropic structures are proportional to perceived differences in color [PP99].

Full tensor information can be represented by diffusion ellipsoids [PJB+96]. Laidlaw et al. [LAKR98] mention as a disadvantage visual cluttering and that small tensors lead to sparsely spaced icons which results in the connection between values being lost. As an improvement the authors normalise the ellipsoids such that their largest radii are equal. Note, however, that this way information about the magnitude of the diffusion is lost. The authors report that rendering regularly spaced and closely arranged normalised ellipsoids achieves a texture like impression.

Westin et al. explain that even with illumination the exact shape of tensor ellipsoids is hard to distinguish and instead propose a new tensor icon consisting of a sphere, a disk, and a rod with a common centre and diameters given by the minimum, medium, and maximum diffusivities, respectively. Different colours are used to represent linear, planar, and spherical cases [WMK+99]. Wiegell et al. [WLW00] represent a diffusion tensor with an octagonal cylinder since it allows better perception of the 3D geometry and better differentiation between linear and planar isotropic diffusion. Poupon et al. use small normalised cylinders [PMF+98].

An interesting visualization of DTI slice images has been developed by Laidlaw et al. using concepts from oil painting [LAKR98]. The projection of the principal diffusion direction onto the image plane is encoded by the stroke direction and the out-of-plane component by the saturation of the red colour component. Diffusion anisotropy is represented by the length/width ratio and transparency of brush strokes and the magnitude of the diffusion rate by the stroke texture frequency. Additional information is given using the lightness of the underpainting and an underlying checkerboard spacing.

Kindlemann et al. [KW99, KWH00] visualize the 3D geometry of the diffusion
3.3 The Human Brain

tensor field using a direct volume rendering technique with the color, lighting and opacity assignment governed by the underlying tensor field. Colours are determined by transforming a constant input vector with the tensor and by using the result to index a Hue-ball which is a 2D spherical colour map. The hue of the resulting colour reflects the principal diffusion direction and the saturation the diffusion anisotropy. Fiber tracts show up as regions of slowly varying saturated colour. The illumination is determined by Lit-tensors which provide a lighting model incorporating Phong illumination for surfaces and illuminated field lines. The opacity assignment is based on the two-dimensional barycentric space of anisotropies defined by equation 3.3-3.5 and is used to select structures according to the type of anisotropy within them. The resulting images resemble the ones obtained by standard direct volume rendering (subsection 4.6.6) except that colour and illumination reveal tissue properties by encoding diffusion direction and anisotropy rather than tissue density and density gradient (e.g., CT). The main disadvantages are the lack of interactivity and the difficulties in discerning edges due to the illumination definition.

Tracking and Visualization of Nerve Fiber Tracts

Over the past couple of years an increasing number of researchers has investigated the tracking and visualization of nerve fiber tracts from diffusion tensor data. Xue et al. track nerve fibers by propagating a line from the centre of a voxel along the direction of the maximum eigenvector until it exits the voxel [XvC’99]. The procedure is continued at the entry point of the next voxel until the inner product of the vector with the vector of the three neighbouring voxels is smaller than 0.75. The authors specify the starting points by manually identifying white matter regions in $T_2$ weighted image slices after which a group of voxels is defined.

Lazar et al. [LWT’03] propose a new algorithm called tensor deflection which uses the entire tensor to deflect the estimated fiber trajectory rather than just using the maximum eigenvector. The authors show that in simulations the deflection term is less sensitive to image noise than the major eigenvector.

Poupon et al. [PMF’98] track white matter fibers using a Markovian model and the assumption that fiber tracks can not end in white matter. In a later paper the authors use knowledge of the low curvature of most fascicles and track them using a bending-energy minimising scheme [PCF’00]. Weinstein et al. [WKL’99] track white matter fiber tracts using an advection-diffusion model which according to the authors gives better results in regions of local complexity where the diffusion tensor data is influenced by multiple features. Hahn et al. [HPP’01] diagonalize the tensor field and bilinearly interpolate the corresponding direction field. Basser et al. [BPP’00] compute the fiber tract trajectories by solving a Frenet equation. Finally Tuch et al. [TRW’02] resolve multiple fiber orientations within a voxel using a diffusion gradient sampling scheme. An additional technique is presented in [WMK’99].

Zhang et al. [ZCML’00b, ZDL’03] introduce streamtubes and streamsurfaces for visualizing DTI data. The trajectory of a streamtube follows the maximum diffusion direction whereas its ellipsoidal cross section represents the medium and minimum diffusivities. The authors normalise the cross section so that it has a constant maxi-
mum diameter and its aspect ratio reflects the aspect ratio of the transverse diffusivities. Streamtubes are initially constructed for each voxel exceeding an anisotropy threshold and are then culled to a representative subset taking into account length, average anisotropy and similarity to neighbouring streamlines. The streamtubes are colour coded with the diffusion anisotropy.

Streamsurfaces identify structures where diffusion occurs predominantly within a plane. The term streamsurfaces used by the author is misleading since they are not related to streamsurfaces common in CFD (see section 4.8.3). Instead the authors define streamsurfaces as integral surfaces perpendicular to the minimum diffusion direction. The surfaces are coloured with the diffusion anisotropy. Interpretation of the visualizations is improved by inserting isosurfaces representing the eyes, ventricles, and the inside skull surface as anatomical landmarks.

Basser et al. [BPP+00] launch fiber tracts as bundles of dense fibers starting from regions of coherently organised white matter. The resulting tracts are then rendered either by surface shading or as density maps where the intensity of an image pixel depends on the number of fiber tracts projecting onto that pixel. Parker et al. [PWKB02] use level set theory and fast marching methods to identify and visualize fiber tracts. Brun et al. [BPKW03] enhance the perception of fiber bundles and connectivity in the brain by colouring fibers according to their origin using Laplacian eigenmaps. Zhang et al. improve the interpretation of the data by using an immersive virtual environment [ZDK+01].

3.3.5 FE Model for a DTI Data Set of the Human Brain

We obtained a diffusion tensor imaging data set of a healthy human brain from Dr. Peter J. Basser and Dr. Carlo Pierpaoli from the National Institute of Health, Bethesda, Maryland, USA. The DTI data set has $128 \times 128 \times 33$ sample points arranged in a regular grid. The slice resolution is $1.72\,mm$ and the distance between slices is $3.5\,mm$. The diffusion tensor field is reconstructed by trilinearly interpolating the components of a tensor over a grid cell. The domain of the data set is a single trilinear element which represents the bounding box of the DTI data set. We use two versions the data set: an unsmoothed version and a version smoothed using a method described in [AB99]. The smoothed data set is visualized in chapter 7.
Technological advances over the past decade have enabled scientists to create ever larger and more complex scientific data sets. Examples are advanced numerical simulations, satellite measurements and medical imaging data. As a result it has become increasingly difficult to understand, analyse and communicate the resulting data sets. Ware et al. summarize this situation by the observation that knowledge has risen exponentially while human capacity to absorb information has stayed constant [WCG00].

Consequently new ways must be found to maximise the information acquired from data while minimising the cost of interacting with it. Scientific visualization is an attempt to achieve this goal by transforming numeric scientific data into one or more images (the visualization) that, when presented to a human observer, convey insight or understanding of the data. The process is often described by means of a visualization pipeline that maps data into graphical representations which are then perceived and interpreted by the user. Scientific visualization improves the perception of features and patterns in the data, facilitates the navigation through and interaction with complex and disparate sets of data and improves the communication of scientific results with peers and the wider community. Furthermore scientific visualization is increasingly integrated into simulation processes. Computational steering of a simulation allows immediate feedback of changes in a simulation process upon parameter change [Mv95] and in combination with visualization techniques helps to recognise errors in computational models early [JP94, JPH’99, JPW00, JPWH02].

This chapter reviews the current state of the art of scientific visualization with a particular emphasis on tensor field visualization. The first two sections give an overview of challenges encountered when visualizing multidimensional data and present a new schema representing the visualization process. The next section gives an overview of data transformation techniques which are used to generate intermediate data representations more suitable for the visualization process. A summary of perceptual concepts and results from cognitive science is followed by a new clas-
sification of visual attributes according to representational accuracy, perceptual dimension and spatial requirements. The classification is used to obtain values for the information content and information density of an attribute.

The main part of this chapter is an extensive survey of visualization techniques for scalar, vector, and tensor fields and is followed by a classification of these techniques. We conclude with a summary of issues relevant for combining multiple visualization techniques and for creating effective visualizations.

4.1 Challenges in the Visualization of Multidimensional Data

The biggest challenge in visualizing multidimensional data is that the visualization must be displayed on a two-dimensional screen (with limited resolution) using colour as an extra output dimension. Animating a visualization introduces an additional output dimension which is frequently reserved for the independent time variable. Using the characteristics of human visual perception it is possible to obtain additional perceptual output dimensions. A third spatial dimension can be perceived if stereoscopic techniques are used or pictorial cues for depth perception are employed (see subsection 4.4.3). Several authors refer to texture as an independent output dimension which makes use of the brain’s ability to identify patterns. In addition the perceptual colour space is three-dimensional though limitations in human colour vision restrict its full usage and usually only two perceptual output dimensions, such as hue and lightness or hue and saturation, are employed. Some authors refer to colour and transparency as independent output dimensions, but since transparency impacts the perceived brightness and saturation of a colour we do not follow this interpretation. In the following we use the term output dimension to refer to both physical (e.g., 2D dimension of the computer screen) and perceptual dimensions such as a perceived depth value.

Rather than creating new graphical representations for each data set it is convenient to use predefined graphical entities (visualization icons) that can be used to represent selected variables (fields) of the data such as temperature and velocity. Each visualization icon employs one or more visual attributes such as colour, shape and texture. The challenge in scientific visualization is to find appropriate icons for a particular visualization task and to combine the visualization icons such that information is enhanced rather than obscured.

In order to develop suitable guidelines for visualizing data the next section presents a visualization schema which relates the visualization process to the perception and understanding of the visualization.
4.2 A Visualization Schema

A multidimensional data set $L^n_m$ consists of $m$ independent variables representing the data domain and $n$ dependent variables defined over the domain [WLG97]. In most applications the independent variables define a two or three dimensional spatial domain and the data set is simply called 2D and 3D data, respectively. An additional independent variable can be introduced by considering time. Both dependent and independent variables can be either discrete or continuous and can have a finite or an infinite range of values. Common examples for dependent variables are scalar fields ($n = 1$) such as temperature, vector fields ($n = 3$) such as velocity, and symmetric tensor fields ($n = 6$) such as stress. Many scientific data sets consist of multiple fields defined over the same domain resulting in a high-order space of dependent variables.

As an example consider a three dimensional finite element data set of a linear elastic solid (see section 2.3). The data set consists of three independent variables defining a sample grid and 15 dependent variables for the stress and strain tensor (6 variables each) and the displacement vector (3 variables). All variables are discrete and finite.

Traditionally the visualization process has been represented by a pipeline [HM90] which performs data encoding. However, already Bertin observed [Ber81] “a graphic is never an end in itself; it is a moment in the process of decision-making”. This means that an effective visualization can only be created if the perception and interpretation of the data is taken into account. As a consequence we extend the traditional pipeline model by a data decoding step as shown in figure 4.1 [WL01a].

The first stage of the data encoding step is the data transformation stage that converts raw data into a form more suitable for visualization. This can involve resampling, data type changes, subset creation, and the derivation of new quantities. The subsequent visualization mapping converts the raw data into a number of visualization icons which represent one or more variables over the whole or a subset of the domain. A visualization icon which represents only one data point is called a glyph.

Some authors (e.g., [Chi00, WCG00]) prefer to subdivide the mapping stage further into visual transformation (or data modelling) and visual mapping. However, in the field of scientific visualization the visual transformation step is usually implicitly performed by choosing a particular visualization icon. For example, a streamline can be considered as a data model connecting points according to an underlying vector field. The visual mapping process does then consist of specifying appropriate parameters for the streamline (e.g., colour mapping a scalar field onto it). Since icons are usually associated with a set of inherent parameters we think that choosing an icon and specifying its parameters are one and the same step which is reflected in our visualization schema.

The rendering stage displays the visualization either on a screen or by printing. Creation of real physical models using texture photomapping has also been reported [CB97]. Note that the display of images using different output media is a complex
problem involving issues such as resampling and conversion between colour gamuts or halftoning [Poy00, SBS99] which are outside the scope of this thesis. Furthermore icons can in general be displayed by different rendering algorithms (e.g., ray tracing vs. hardware supported polygon scan conversion) the choice of which is usually dependent on the required image quality and performance considerations (e.g., interactivity).

**Figure 4.1. A visualization schema.**

The data decoding step describes how visual information is perceived and processed and consists of visual perception and cognition. Visual perception is the translation of a visual stimulus into a mental image of the scientific data whereas cognition refers to the interpretation and understanding of the data.

A visualization is represented by visual attributes such as geometry, position, colour, and textual attributes such as text and symbols which themselves are represented by simple visual attributes. The visualization is effective if the decoding can be performed efficiently and correctly. “Correctly” means that perceived data quantities and relationships between data reflect the actual data. “Efficiently” means that a maximum amount of information is perceived in a minimal time. An interesting observation is that visual attributes are processed in the right half of the brain whereas text and symbols are processed in the left half of the brain [Edw00]. Using both types of attributes appropriately might maximize the amount of perceived information.

The visualization scheme can be extended further by interpreting it as a bidirectional model as has been suggested by Groß [Gro94a]. This interpretation complements the visualization pipeline (⇒) with an interaction pipeline (⇐) in the reverse direction. The interaction pipeline allows the user to retrieve the original data from a pixel of the displayed image and enables the integration of physically based modeling and visualization under real-time conditions.

Various researchers have tried to employ other sensory organs than the eye to perceptualize data in order to avoid visual overload. Grinstein et al. experimented with the sonification of scientific data employing the loudness, pitch and orchestration of sound as additional output dimension [GS90, GL95]. Corresponding tools such as sound probes and property sonification have been proposed [SMP98]. Sonification has also been employed to perceptualize uncertainty [LWS96] and volume images [RN96]. Other experiments have utilized tactile perception as an output dimension [HF96].
Summarizing we conclude that two issues are vital for creating effective visualizations: The data must be transformed and mapped onto meaningful graphical entities (visualization icons) which can be easily perceived and interpreted. An understanding of the human visual perception is important for the successful design of these icons and in order to create an effective visualization from a selection of visualization icons and contextual cues. We explore these topic in the following sections.

4.3 Data Transformation

Scientific data can consist of multiple scalar, vector, and tensor fields. The previous section introduced a range of visual attributes available for encoding such data. In order to map data onto visual attributes it is often necessary to convert the data first into a more suitable form. This is achieved by using a data transformation which modifies the independent variables, the dependent variables or both.

4.3.1 Transformations Based on the Independent Variables

In general the independent variables of a scientific data set define the spatial and the temporal domain. Transforming data by considering a part of this domain only, e.g., a slice in an MRI data set, is easy and intuitive. This type of transformation is often called *subspace selection* and is an example of a data reduction. Subspace selection is frequently performed interactively using slicing tools in order to choose individual space or time slices, selection tools to define a region of interest, or more exotic tools such as magic lenses [FG98]. In addition semantic information such object boundaries can be used to define interesting subspaces. For example, Max et al. achieve good results by restricting a flow visualization to the vicinity of contour surfaces [MCG94].

Transforming data by considering the independent variable often improves the perception of the data set by uncovering hidden features and hence facilitates the exploration of the data set (see subsection 4.11.3). However, the understanding of the data set might not be enhanced since structural information contained in the dependent variables is not extracted.

4.3.2 Transformations Based on all Variables

Many scientific data sets are given by discrete sample values and must be transformed into continuous data before transformation. This type of data transformation is an example of data enrichment and can be accomplished by using interpolation techniques such as scattered data interpolation [HL92a] or finite element interpolation previously described in subsection 2.4.2. A survey and analysis of different interpolation techniques is given in [WT04]. When dealing with 3D scalar data the
problem is often referred to as volume reconstruction. Optimal reconstruction functions for specific application areas such as volume rendering are an ongoing topic of research (e.g., [ML94, MZ95, MMK+98, MMC99]). Reconstruction of vector and tensor data is more complicated. A technique based on spectral analysis has been suggested by Aldroubi and Basser [AB99].

Sometimes discrete data is obtained from continuous or other discrete data by sampling. The conversion between two discrete data sets is termed resampling and often involves the use of interpolation techniques. Discretizing data and applying image processing techniques can be utilized in scientific visualization to reveal hidden information [KK93].

Several more advanced data reduction and simplification techniques exist. The general aim is to find a balance between data size and data quality. Advantages are less storage, faster visualization, easier placement of visualization icons and extraction of visual features.

An example is given by Leeuw and Post [dP94] who use statistical methods such as the mean and standard derivation in order to reduce vector data in accordance with its distribution characteristic. Vector data has also been simplified using clustering and principal component analysis by employing an error metric based on streamline distance [HWHJ99] or vector similarity [Tv99].

A generalised data simplification process can be obtained by using the concept of data signatures [WFL+00]. Data signatures represent data sets by numerical entities (e.g., vectors) and can be used to measure the difference between two data sets using an appropriate metric. Construction of data signatures is based on such diverse measures as velocity gradient tensors, critical points and their eigenvalues, covariance matrices, intensity histograms, content segmentation and conditional probability.

A simple simplification technique is to represent subsets of data, such as time slices, by statistical properties such as the median, maximum and minimum value. This technique forms the basis of box plots used in charting [Cle93].

Classical statistics offers many more techniques for reducing the dimension and the amount of data. An important subset of statistical analysis algorithms are methods designed for multivariate data (multivariate data analysis) which can be defined as a set of entities where each element e_i consists of a vector of n observations (x_{i1},...,x_{in}) [War94]. In contrast to scientific data multivariate data does not have the necessary semantic information to allow a differentiation into dependent and independent variables. An example is survey data classifying m people (cases or rows of data) according to n personal preferences (variables or columns of data).

The following techniques are frequently used for dimension reduction: Principal component analysis represents data as a set of new uncorrelated variables which are linear combinations of the original variables and are ordered by importance. A low dimensional overview associates a Euclidean distance metric with the dissimilarity between any two data points and therefore allows a mapping into two dimensional space. Multidimensional scaling combines dimensions with a high correlation of data and attempts to find a linear combination of original dimensions with a maximal spread of data. Care has to be taken when applying these techniques to data rep-
4.3 Data Transformation

represented by a set of independent and dependent variables [CC80]. Other statistical techniques aim to reduce the overall amount of data. These include cluster analysis and vector quantization which attempt to find clusters of data with high a correlation [Gro94b].

A more general approach to find structures and correlated variables in a data set is to use data mining techniques [Rhy00]. In addition to the methodologies from classical statistics data mining also employs artificial intelligence techniques such as Kohonen maps and learning vector quantization [Gro94b]. Recently several methods for visual data mining have been suggested where user-interaction guides projection [HKW99, KS02] and clustering-based [RKJH99] data mining techniques and replaces automated decision making. Surveys of visual data mining techniques are given in [Kei02, dL03].

4.3.3 Transformations based on the Dependent Variable

Finally data can be transformed by considering the dependent variable only. Advantages of this approach are that frequently intuitive results can be derived, a usually easy implementation and that structure not apparent in the original data can be uncovered. Transformations of the dependent variable can be represented as mathematical operators. Techniques are classified into data reduction, data expansion, or data modification algorithms, depending on whether the dimension of the dependent variable reduces, increases or stays the same, respectively.

Examples for data reduction techniques are the computation of vector magnitude, eigenvalues and eigenvectors. Other applicable operators include the dot product, matrix determinant, evaluating surface curvature and distance metrics [SML96a]. Projections can be used to reduce the dimension of the dependent variables. Max et al. suggest different projections of 3D vector fields onto curved surfaces [MCG94].

Data expansion is commonly obtained using a gradient operator for scalars and a Jacobian for vector data. Both of these operators give neighbourhood information at a point and can be utilized to detect local features in a data set (e.g., extrema, ridges).

Finally examples of data modification are coordinate transformations and various techniques for vector field and tensor decomposition. For instance, Polthier [PP00] and Tong et al. [TLHD03] decompose a vector field into a divergence-free part, a curl-free part and a harmonic part. Lavin et al. [LLH97] decompose a tensor $T$ into a deviator tensor $D$ and an isotropic tensor $U$ where

\[
U = \frac{1}{3} q I \\
D = T - U
\]
I is the identity matrix and

\[ q = \sum_{i=1}^{3} \lambda_i = \sum_{i=1}^{3} T_{ii} \]

is the first tensor invariant which is identical to the sum of the eigenvalues of the tensor (see subsection 2.1.1). Visualizing the deviator only removes the constant contribution of the isotropic component which otherwise could obscure structure in the visualization [BP98b, Bor98].

Many applications require more specialised derived measures. In subsection 3.2.5 and 3.2.7 we introduced specialised measures derived from the strain tensor and in subsection 3.3.3 we presented a large variety of measures characterizing the diffusion tensor.

4.4 Human Visual Perception

The visualization schema previously shown in figure 4.1 indicates the importance of understanding human visual perception in order to create effective visualizations. Visual Perception is the interpretation of visual stimuli by the brain and is determined by both the physiology and psychology of the human visual system. This section first reviews the anatomy and physiology of the visual system and then gives an introduction into low and high-level perceptual processes with a separate subsection dedicated to colour perception. The section concludes with a classification of visual attributes.

4.4.1 Anatomy and Physiology of the Eye

The human eye consists of an optical system, represented by the cornea, the lens and the iris, and a neural system of which the main component is the retina [Fer98]. The optical system is responsible for refracting and focusing the incoming light. The neural system contains photo receptors and associated neural tissue and converts incoming light within the visible spectrum (380-760nm) into nerve signals. Two classes of photo receptors are found on the retina: Rods are highly sensitive to light and are responsible for achromatic vision. Cones enable colour vision and are divided into three types whose spectral response function have their peak for red, green, and blue light, respectively. There is a significant overlap between the spectral response functions and each spectral response function is highly nonlinear [FvFH92]. According to the opponent-process theory a subsequent layer of ganglion cells processes colour stimuli into three antagonistic responses: red-green, blue-yellow and white-black [Sch96]. The theory explains several colour illusions such as simultaneous contrast and after-images explained in subsection 4.4.4 [Wan99].

After being generated in the neural system of the eye visual stimuli travel via different visual pathways towards the visual cortex. The primary pathway consists of the optic nerves, which cross at the optic chiasm and project to the visual cortex at the posterior side of the brain [Fer98]. This pathway seems to be responsible for
the “what” information [Dav91]. It has been suggested that the primary pathway is further divided into separate pathways for visual processing of shape, colour, movement and depth [LH87] and that a secondary pathway exists in the lateral regions of each hemisphere which seems to be responsible for the “where” information [Dav91].

Visual information is processed in different layers of the visual cortex [Fer98]. Low level visual information is processed in “layer 4B” by three types of nerve cells, which are sensitive to colour and contrast, edges, and orientation and direction of movement. It has been suggested that maximum intensity information is lost during processing and only contrast information is transfered along later stages of visual processing [Fer98]. Other layers are responsible for higher level function such as stereoscopic depth perception. Current research indicates that further visual processing is divided into two functional layers responsible for localization and recognition of function.

The visual system possesses several fundamental characteristics which determine its functionality. *Adaption* refers to the eye’s capability to adjust to different lighting condition. Full adaption to a dimly illuminated environment can take 15-30 minutes; adaption to a brightly illuminated environment takes only a couple of seconds [Sch96].

*Visual Acuity* describes the minimum perceivable separation between the centers of adjacent dark and light bars. Under optimal conditions visual acuity is about 0.5 minute of arc [HIR99] though the ability to localize an object in the visual field is higher by a factor of 5 [SL82]. Spatial acuity increases with luminance [Fer98] and changes non-linearly with contrast [SL82]. Spatial resolution by colour differences is only half of that achieved by luminance contrast [Dav91].

Finally the human eye has the ability to fuse rapidly interrupted images. The temporal resolution of the human eye is less than 16 frames/sec so that higher frame rates are usually perceived as continuous motion. However, the eye is sensitive to flicker up to 60Hz [GL95].

### 4.4.2 Visual Processing

The perception of a scene is the result of two major processing stages [Sch96]. The initial *preattentive stage* allows perception of very simple features without conscious attention. An example is the instantaneous perception of a red dot in a cloud of blue ones. The second principal visual processing step is the *focused attention stage* which involves conscious examination of a scene, rapid mental calculations and quantitative reasoning. This stage is responsible for identifying complex unitary objects and complex quantitative information.

Preattentive vision seems to be dependent on primitive textural features (*textons*) such as length, width and orientation of simple elongated shapes as well their end connections, angle orientations, and intersections [Sch96]. Apart from these preattentive vision also exists for shape, curvature, closure, colour (hue), intensity and more complex visual attributes such as texture and depth [HIR99, Hum92].
The underlying mechanism has been attributed to humans' having different sensory dimensions for basic visual attributes such that a unique feature in any dimension is immediately detected [Dav91]. This might explain why combinations of preattentive features usually do not work [HIR99].

Preattentive vision can be suppressed by cognitively demanding tasks, the presence of visually important features or where the preattentive feature is not task relevant. Preattentive vision applies not only to feature detection but also to region segregation and basic quantitative tasks such as estimating the percentage of red dots in a cloud of red and blue dots [HIR99]. Research indicates that preattentive features are ordered by importance with hue being the most important especially in dynamic environments [HIR99].

### 4.4.3 Visual Attributes

The previous subsection described the basic organisation, properties and processes of the visual system. This subsection explains how perception of more complex forms and concepts is achieved by processing visual stimuli into integrated units through several layers of perceptual mechanisms.

The most basic integrated units are the visual attributes colour, line orientation [GL95], and contrast which is fundamental for the perception of contours [Sch96]. Other basic perceptually significant attributes are transparency, position and size [ERS'99, Fer98, Dav91]. All of these attributes are determined in the first initial visual processing step performed by the brain.

The brain utilizes the previously listed low-level visual attributes for performing more complex visual tasks such as the perception of shape, Gestalt, and depth which together are referred to as spatial vision [Fer98]. Other higher order tasks are figure-ground perception and texture perception. The attributes perceived by these tasks are called high-level visual attributes.

Texture is perceptually characterized by its spatial frequency, contrast and orientation [Sch96, WK95]. Recognition of feature patterns is accomplished using primitive textural features (textons) such as length, width and orientation with line segment orientation being particularly important for visual segmentation of surface textures [PGLS95]. Note that textons are also preattentive features. Pattern detection is orientation dependent and is influenced by adaption (familiarity) [Fer98].

Shape information is directly derived from luminance, motion, binocular disparity, colour, and texture, with luminance yielding shadow and subjective (illusory) contour information [Dav91]. Shape perception is dominated by the curvature of the silhouette contour (figure-ground boundary) and 3D surface shading [Hum92]. Shape perception is highly orientation dependent: Rotated versions of the same form can sometimes be perceived as different shapes. Perception can also be dependent on previous stimuli [Sch96]. Familiar shapes and configurations can improve the recognition of a target if they are a part of it [Sch96].

Depth perception is achieved using binocular vision and visual cues [IIC]. Binocular vision includes disparity, accommodation and convergence and motion parallax.
Binocular disparity is the most important depth cue for medium viewing distances and refers to the fact that the retinal images of an object viewed by two eyes which are slightly displaced differ slightly. The displacement of the retinal images of an object is converted by the brain to depth information. Accommodation is the adjustment of the focal lengths of the lens and convergence is the angle made by the two viewing axes of a pair of eyes. These cues are only effective for small distances and in combination with other binocular cues. Motion parallax is the effect that the relative distance an object moves determines the amount its image moves on the retina. For visualization purposes binocular vision is achieved by using stereo goggles or VR Head Displays.

Independently of binocular vision, visual cues such as retinal image size, brightness, shade and shadows, linear perspective, overlay, texture gradient, and aerial perspective [Hum92, Lip] can be used to aid depth perception. Aerial perspective stems from the observation that colours on the horizon usually appear bluish blurred. As a consequence the brain associates such colours with large distances. Visual cues are also called *psychological cues* since they are learned, i.e., they are assisted by experience.

The concept of *Gestalt* originates from the fine arts and expresses the notion that the “whole contains more information than the parts” [Edw00]. An example is the perception of a circular arrangement of symbols as a circle. Perception of Gestalt is influenced by proximity, similarity, continuation, closure, symmetry, and the law of Prägnanz, which states the the eyes tend to see the simplest and most stable figure [SL82, Sch96]. Context might also play a role in Gestalt perception [Hum92].

Figure-ground perception describes the observation that an object can be instantly separated perceptually from its background [Sch96]. This is due to usually physically different attributes of a figure and its background but is also influenced by size, angle, and association with meaningful shapes [Sch96].

The previously described perceptual processes are all utilized for object recognition. Other important factors are stored knowledge acquired by perception and conception of sensory information. Object knowledge is divided into functional and descriptive information [Dav91]. It has been proposed that 3D object recognition is achieved by mentally decomposing an object into 24 perceptual building blocks and by identifying their size, orientation and position [Sch96]. The visual system is especially adapted to some specialised classes of objects such as facial expressions. For instance, Chernoff utilizes the brain’s ability to quickly recognize small changes in facial expressions for the design of *Chernoff faces* for information visualization [Che73]. Note however that facial expressions are not recognised preattentively [MER99]. Understanding the process of object recognition is important when designing visualization icons.

As previously indicated in section 4.2 visual information is predominantly processed in the right hemisphere of the brain whereas the left hemisphere is responsible for recognizing symbols and words. Word recognition includes basic feature detection on the lowest visual processing level but largely involves higher order processing tasks such as detection of letters, spelling patterns, syllables, and phonological codes.
which are independent of the visual system [SL82]. These observations might be one reason why careful addition of text and symbols can improve the perception of a visualization. Overuse of textual information must be avoided, though, since it leads to visual cluttering and information obstruction [Tuf83].

Some additional remarks about colour vision are given in the next subsection.

4.4.4 Colour Perception and Colour Spaces

Although colour is a basic visual attribute its perception by the brain is a complex process. As explained in subsection 4.4.1 colour is perceived through stimulation of the visual system by electromagnetic waves within the visible spectrum (380-760nm). A colour sensation is the result of a stimulus of three types of colour sensors (cones) on the retina with peak responses for red, green and blue light [FvFH92]. Different spectral distributions can produce the same colour sensation and are called metamers. Encoding the light spectrum into amounts of light absorbed by the three types of cones forms the basis of trichromacy, i.e., human colour vision is three-dimensional. Using the concept of trichromacy a perceived colour can be represented as weighted sum of three primary colours red, green, and blue. The resulting RGB colour space is represented by a unit cube with black at the origin, white at the position (1,1,1) and the primary colours at the end of each axis aligned edge. Figure 4.2 (a) shows an image of the RGB cube looking along its main diagonal from white to black.

The RGB colour space is used to represent colours on a computer screen and models the physiology of colour perception. The appearance of colour, however, is also influenced by perceptual and cognitive effects, i.e., the processing of colour stimuli and our knowledge of the environment [Sto00]. A more intuitive representation of colour is given by the perceptual attributes hue, brightness and saturation which reflect the physical attributes wavelength, intensity, and spectral purity, respectively. A hue is in general determined by the dominant wavelength of a spectral distribution, with short wavelengths being perceived as blue and long wavelengths as red. Brightness varies with the physical intensity of light, but some hues such as yellow appear brighter than others such as blue even if they have the same intensity [Sch96]. Saturation reflects the relative amount of a hue and is hence related to spectral purity. Monochromatic light, which consists of only one wavelength, is perceived as highly saturated whereas all shades of grey have a saturation of zero.

The colour space corresponding to this more perceptual classification is termed HSB (hue, saturation, brightness) space. The HSB colour space is reflected in the HSV (hue, saturation, value) and HLS (hue, lightness, saturation) colour models for raster graphics [FvFH92]. An image of the HSV colour space is shown in figure 4.2. Note that some authors use the term lightness to indicate the luminance and the term brightness to indicate the perceived intensity.

The CIE colour space reflects both perceptual and physiological issues. Colours are constructed as a mixture of three special “supersaturated” primary colours that do not correspond to real colours but that have the properties that all visible colours
4.4 Human Visual Perception

Figure 4.2. Images of the RGB (a), HSV (b) and CIE (c) colour spaces.

can be represented as a positive combination of them. One of these primary colours models the response of the eye to constant luminance (luminous-efficiency function). Thus colourfulness is separated from brightness [Sto00]. Figure 4.2 (c) shows the CIE chromaticity diagram. The horseshoe shaped area contains all visible colours with maximum luminance. Connecting the horseshoe shaped boundary to the origin, which represents black, defines the CIE colour space of all visible colours. The coloured triangle in figure 4.2 (c) represents the monitor gamut, i.e., all colours (with maximum luminance) which are obtained by linear combinations of the three primary colours of the monitor. Using a non-linear transformation the CIE colour space can be converted into the CIE LUV or CIE Lab colour space which are both additionally perceptually uniform, i.e., perceived colour differences vary linearly [FvFH92].

Although colour is represented by a 3D space the brain’s ability to locate a colour in this space is limited. Davidoff reports that even colour experts have difficulties separating hue and lightness [Dav91]. Colour perception is also dependent on the wavelength with long wavelengths perceived most easily whereas short wavelengths are only identified where luminance is relative high. Changes in saturation with constant lightness and hue are perceived most easily at the extremes of the spectrum [FvFH92].

Colour perception is influenced by surrounding colours (colour illusions). For example, a colour appears less saturated against a dark background than against a light background and a patch’s colour is perceptually shifted by the colour of adjacent patches (simultaneous contrast). If colours of different intensities meet, non-existing intensity changes (Mach bands) are perceived [KK93]. Prolonged exposure to a colour can also change perception of subsequent visual impressions since it produces an afterimage of the complementary colour [Sch96]. In addition some black and white patterns can cause colour sensations (subjective colours) [Sch96].

Often colours are mentally associated with certain properties or functions. For example, blue and violet are generally associated with “cold”, red and yellow with “hot”, high saturations with “high density”, lighter colours with “more active” and
finally red is frequently associated with “critical/dangerous” and green with “safe values” [Dav91]. Note that such associations are dependent on the environment (e.g., cultural influences). For example, in the Chinese culture red is usually associated with “good luck”.

4.4.5 Classification of Visual Attributes

Not all visual attributes are equally well suited for displaying quantitative information. The perceived scale of many attributes is a power of their actual scale (Steven’s law) [Cle85]. The power is close to one for the perception of length so that length variations can be estimated quite accurately. For area and volume changes the power is smaller than one so that small areas are usually perceived larger than they actually are and vice versa for large areas. In addition perception of visual attributes can be influenced by orientation, e.g., angles with a horizontal bisector are seen as larger than angles with a vertical one [Cle85]. Also it has been shown that slope changes influence the perception of vertical distances.

In this section we suggest a classification of visual attributes according to representational accuracy, perceptual dimension and spatial requirement [Wün04b]. The classification supports the identification of suitable visual attributes for representing a given data set and hence forms the basis for mapping data onto visualization icons. An alternative classification of visual attributes for information visualization has been suggested by Dastani [Das02] who classifies visual attributes into spatial attributes, non-spatial attributes and topological attributes and derives from that different perceptual structures. The aim is to find a structure-preserving mapping from data structures onto perceptual structures.

We introduce the term representational accuracy as a measure of how accurately a human can estimate a quantitative variable represented by a visual attribute. Cleveland shows that a quantitative variable is most accurately represented by a position along a scale, and then in decreasing order of accuracy by interval length, slope or angle, area, volume and colour as indicated in the following diagram [Cle85].

- highest accuracy of representation (position on scale)
- interval length
- slope/angle
- area
- volume
- colour

The above ranking changes when visualizing ordinal or nominal data which occurs frequently in information visualization [Mac86].

Since more complex visual attributes are based on basic visual attributes the suitability of a high-level visual attribute for a visualization task depends on the low-level attributes dominating its perception. For example, a spot noise texture consists of “smeared dots” with a length and a direction [van91]. The texture is
therefore well suited for representing 2D vector fields or 3D vector fields which are constrained to a surface.

We differentiate visual attributes further by their perceptual dimension and spatial requirements. The perceptual dimension of a visual attribute refers to the number of its dimensions which can be perceptually differentiated. Length and slope represent only one dimension and the direction of a 3D vector represents two independent dimensions. As mentioned previously, colour, while being represented as a 3D space, has only 2 easily distinguishable perceptual dimensions, e.g., hue and brightness. Texture is usually composed of several basic visual attributes such as colour and the length and orientation of texture elements. The perceptual dimension of a texture is therefore in the ideal case the sum of the perceptual dimensions of the inherent (independent) basic attributes. An additional perceptual dimension is given by the spatial frequency of a texture. Similarly shape has been shown to represent multiple independent perceptual dimensions.

We define the spatial requirement of a visual attribute as the smallest unit of space (i.e., pixels on a screen) necessary to identify a piece of information. A point has a dimension of zero, a line a dimension of one, etc. Note, however, that a point on the screen is represented by a pixel which occupies a finite screen area. Analysing various visual attributes we can see that colour has a minimal spatial requirement only limited by the resolution of the human visual system whereas a texture requires a much larger space on the output medium to enable the viewer to identify inherent information. For example, a pixel of a spot noise texture contains no information since neither direction nor length of the represented vector field are apparent.

Using this classification we define the information content of a visual attribute as the product of representational accuracy and perceptual dimension. The information density is given by dividing the information content of a visual attribute by its spatial requirement.

A listing of common visual attributes classified using the above criteria is shown in table 4.1. For clarity the spatial requirement is described both subjectively (low-high) and then in brackets more objectively by the dimension of the occupied space. For example, a position on a scale is given by a point (dimension 0, however the scale itself requires some space) whereas volumes and other 3D shapes occupy a three-dimensional region. Colour can be represented by points, but colour perception is very poor for isolated pixels so that for many applications the space requirement of the colour attribute is higher than zero dimensions.

4.5 Visualization Icons

In order to create an effective visualization, scientific data must be mapped to visual attributes in a way that optimizes perception and understanding. The task is difficult since the perception, interpretation and comprehension of visual input is influenced by context, attentional focus, expectations, prior knowledge, past experiences and subjective biases [HIR99].

The visualization task can be facilitated by using standardized visualization
icons for scalar, vector and tensor fields. Visualization icons are graphical objects that encode scientific data by visual attributes. In general the independent variables (specifying the domain) of the scientific data are reflected in the spatial (and temporal) position of an icon leaving colour, textures, shape, orientation and size (length/area/volume) to encode the dependent variables (the values defined over the domain).

The following three sections introduce a wide variety of scalar, vector and tensor icons. Each section concentrates on methods for static 2D and 3D data sets since this type of data has attracted most of the research on visualization techniques and since this type of data is dominant in biomedical science. The choice of an icon is often determined by the dimensionality of the data. For example, vector arrows in 2D fields are unambiguous, but the same is not true in 3D and additional cues are needed. Most of the icons can be applied to time-varying data by changing the icon smoothly over time. Specialised techniques for time-varying data sets are mentioned where appropriate.

### 4.6 Scalar Icons

The simplest type of data sets in scientific visualization are scalar fields which have just one dependent variable. Typical examples are density, temperature, and pressure.

#### 4.6.1 Colour Mapping

A popular way to visualize scalar fields is to associate the field’s range of values with a colour scale. The scalar field is then displayed over a domain by colouring each point of the domain according to its field value. The most frequently used domains for this colour mapping process are lines or surfaces intersecting the visualization domain. The example in figure 4.3 shows that using an appropriate colour scale can reveal otherwise hidden structures. Colour scales can also be mapped onto other visualization icons such as height fields, hyperstreamlines, and textures, in order to increase the encoded information content. Speray proposes volume probes as an
additional application. The probes consist of colour mapped polygons assembled to a trihedral or paddle-wheel shape and they are used to interactively explore data sets [SK90]. A generalisation of colour mapping is used in direct volume rendering where scalar field values are associated with both colour and opacity values using transfer functions (see subsection 4.6.6).

![Figure 4.3](image1.png)

**Figure 4.3.** Density of matter of an astrophysical jet in intergalactic space visualized by using two different colour scales (©1993 IEEE Computer Society Press [KK93]).

Colour mapping is well suited to give an overall impression of the distribution of a scalar field. Note, however, that subsection 4.4.3 showed that quantitative information displayed by colours can not be perceived accurately. Also the effectiveness of the colour mapping process depends on perceptual issues and the colour scale used. Levkowitz and Herman summarize the following desirable properties of a colour scale [LH92]:

- colours should be perceived as preserving the order of the scalar values they represent
- colours should convey the distances between values they represent and should associate related values and separate unrelated values.
- colours should be continuous for a continuous value range.

The scale should additionally accentuate important features while minimizing less important or extraneous details. Artistic or aesthetic quality can also be important. Note that a colour scale can not be judged in isolation since colour illusions such as simultaneous contrast [War88] can reduce its effectiveness (see subsection 4.4.4).
Various colour scales have been presented in the literature [RO86, War88, LH92]. Some examples of the colour scales described in the following paragraphs are illustrated in figure 4.4. A special scale is the linear gray scale which varies linearly from black to white. Levkowitz and Herman report that this scale resulted in a better identification of simulated features in medical images than any of the tested colour scales [LH92]. Gray scales have also been successfully employed in combination with colour scales in order to represent restored missing data. The technique blends missing data into the existing data using luminance interpolation while at the same time allowing the scientist to distinguish the restored data by its lack of colour [TCS94]. Adding a constant hue to a gray scale creates a univariate colour scale. The main disadvantage of gray scales is their limited perceived dynamic range of only 60-90 noticeable values [LH92].

Figure 4.4. Several examples of common colour scales and colour scales described in the literature.

Double ended colour scales are obtained by pasting together two monotonically increasing scales. Their advantage is a clear visual differentiation between low, middle and high values [HIR99]. The rainbow colour scale is defined by the horseshoe shaped boundary of the CIE chromaticity diagram and contains all fully saturated colours. Rheingans and Landreth point out that it is potentially misleading since the brightest colour (yellow) is in the middle [RL95]. Also it has been suggested that this colour scale organises data into discrete regions and might miss details within a colour region [BRT95]. The heated-object scale from Pizer and Zimmermann increases monotonically with brightness from black through red, orange and yellow to white [RL95]. Its main advantage is the association of colours with high and low values (temperatures). The optimal colour scale from Levkowitz and Herman maximizes the number of
distinct perceived colours along the scale and increases monotonically with both brightness and RGB components [LH92]. The authors report that this scale was preferable to the heated object scale when identifying features in medical images.

In many applications specialised colour scales have been developed which make use of the association of colours with field values and properties. Examples are specialised scales for meteorological data [fCG93] where cold and hot temperatures are represented as blue and red, respectively, and snow and blowing dust are coloured white and brown, respectively. Brewer et al. have developed colour schemes for maps which are also suitable for colour blind people [BHH03, Bre].

Other specialised scales are common in biomedical imaging. For example, if flows are imaged using echocardiography then by convention flows towards and away from the transducer are represented in red and blue, respectively [Duk]. Velocity is encoded by the saturation of the hue. The range of perceivable values can be increased by extending the red end of the colour scale to bright yellow and the blue end to bright pale blue.

Several authors have researched how optimal colour scales can be found semi-automatically. A rule-based tool for colour map selection based on the data type, spatial frequency, and the user’s task is presented in [BRT95]. A tool for the dynamic exploration of colour maps is described in [RT90]. Unfortunately these tools are quite limited in their application and it is useful to summarize here some general recommendations for the selection of a colour map.

For shaded curved surfaces colour maps with only hue variations are recommendable since the luminance gradient is used as a shape cue [HIR99]. If simple shapes are used shading can be disabled to encode extra information into the colour scale. Ware suggests that a colour scale which varies in both luminance and hue is well suited to represent both field and surface properties by minimizing simultaneous contrast [War88, HIR99]. Small disturbances in a scalar field can be emphasized by using histogram equalization or by skewing the colour table [KK93, Example 6-1, Example 11-3].

It is also possible to represent two scalar fields simultaneously by colour mapping using bivariate colour scales. Encoding is accomplished by using either two colour components (such as red and green) or two perceptual properties (such as hue and saturation) [HIR99]. Using hue variations for both scales is not recommendable since it hinders the localisation and comparison of values [Tuf83]. Rheingans and Landreth suggest an alternative bivariate colour scale where the two axes express changes in saturation of two complementary colours. The resulting visualization shows an equilibrium of the two scalars as gray values with the magnitude of both scalars encoded as brightness and any imbalance between them shown as hue variations. The colour scale is best suited to display correlation between two variables [RL95].

A discretized colour scale is called a colour band and if used for colour mapping shows isocontours and facilitates the comparison of values in non adjacent regions. Similarly discretizing a bivariate colour scale results in a colour matrix. An example is given by Viecelli and Keller who use a colour matrix to visualize two independent variables by hue and saturation [KK93, p.46]. Instead of discretizing a complete
colour scale it is also possible to introduce sporadic discontinuities in order to highlight important boundaries in data. An example is the separation of a land mass from water [MHC90].

4.6.2 Contours and Isosurfaces

The overall distribution of a three-dimensional scalar field $\rho(x)$ can be indicated by constructing a $c$-isosurface

$$S = \{x | \rho(x) = c\} \quad (4.1)$$

which consists of all points $x$ where the field value $\rho(x)$ equals a given iso value $c$. The surface divides the domain into regions where the scalar field is either greater or smaller than the iso value $c$.

Numerous algorithms (so-called polygonization methods) have been proposed for the efficient computation of isosurfaces (e.g., [WMW86, Blo88, DK91, Wal91, Wv92]). The Marching-Cube algorithm [LC87] has been one of the earliest and most popular methods. The algorithm requires as input a regular grid of sampled field values and “marches” through the volume cell-by-cell. Each grid cell has eight sample values at its corners. The method constructs a tessellation by computing for each cell the intersection points of the cell’s edges with the isosurface and by connecting these intersection points with triangles obtained by a table look-up. The original algorithm suffers from holes due to ambiguities [Diu88]. Various modifications and alternative methods have been proposed [NH91, HH92, GH95]. Ambiguities and topological considerations for isosurface construction are described in [vW94].

Interactive display rates and reduced storage requirements can be achieved by utilizing adaptive methods [BW90, HW90], mesh reduction techniques [KT96, HDD+93] and multi-resolution meshes [CPD+96, Hop97, LDW97]. Shen presents an efficient isosurface extraction for time varying scalar fields [She98]. Multiple isosurfaces for a data set are efficiently computed by preprocessing the data set [Wv92, IK95, LSJ96, CMM+97]. Various polygonization methods are compared in [SHSS00]. Additional information is found in reviews of polygonization methods [NB93, Kal92] and optimization techniques [Wün97].

In many applications semi-transparent isosurfaces are used to reveal inside structure or to display layered surfaces. Interrante et al. improve the shape and depth perception of such surfaces by illustrating them with texture strokes which are advected along the directions of the principal curvatures [IFP96, IFP97]. Similarly principal-direction driven line integral convolution can be employed [Int97]. Surfaces in medical imaging have been enhanced using ridge and valley lines [IFP95]. Instead of using transparencies it is also possible to employ opacity-modulated textures [Rhe96] or to approximate isosurfaces by clouds of oriented particles [van93a, WH94] or small spheres [KK93]. The shape of an isosurface can also be emphasized by colour mapping it with the Gaussian curvature [NFHL91]. DeCarlo et al. [DFRS03] improve shape perception by using suggestive contours which are lines drawn on clearly visible parts of a surface, where true contours would appear with a small variation of the viewpoint.
Isosurfaces can be generalised to *interval sets* \( S = \{ x \mid c_1 \leq \rho(x) \leq c_2 \} \). Guo presents a method to efficiently triangulate interval sets and to render them as semi-transparent surfaces or clouds [Guo95]. Fujishiro et al. use a modified marching cube algorithm to extract an interval set as a solid data structure [FMS95].

Equation 4.1 can also be employed in 2D in order to define (iso)contours. Contour lines can be drawn by tracking [DBV89] or by employing similar techniques as suggested for the 3D case. Saito and Takehashi [ST90] propose an algorithm based on image processing techniques and also introduce curved hatching as a method to achieve an even visual distribution of contour information. Contouring can be considered as a natural extension of colour mapping which constructs explicit boundaries between similarly coloured areas [SML96a].

### 4.6.3 Textures

An increasingly popular technique in scientific visualization is texturing. Similar to colour mapping texturing generates a continuous representation without added geometry. In addition gray-scale textures can be colour mapped in order to encode multiple scalar fields at once.

An early texturing example by Crawfis and Allison uses bump mapping to encode wind velocity by a chaotic roughness of the surface [CA91]. In a more recent work Healey and Enns introduce perceptual texture elements (*pexels*) and examine the use of their height, density and regularity to visualize scalar fields [HE98, HE99]. A later paper attempts to visualize multiple scalar fields and the relationships between them by combining layers of linear texels element with varying luminance and orientation [WEL⁺00].

Texturing is particularly popular in the field of vector field visualization and several techniques will be presented in subsection 4.7.5.

### 4.6.4 Particle Systems

Traditionally particle systems have been used most frequently for the visualization of vector fields. A simple application for scalar fields is the use of randomly positioned colour mapped particles to show the internal distribution of the field [PW95].

Particles can also be used to approximate isosurfaces by moving them along the gradient towards a given isovalue. Particle density is controlled by using birth/death events [PW95] with an even particle distribution being achieved by a repulsion mechanism [WH94]. Crossno and Angel increase particle density with local curvature [CA97]. Shaded surface representations are achieved by considering particles as small surface facets modelled as points with a normal [van93a]. Recently several methods have been presented for the direct rendering of surface models defined by large sets of sample points [RL00, PG01, ABCO⁺03, DVS03].

Pang proposes an expandable visualization environment build on the metaphor of spraying particles in the visualization domain [Pan94, PA94, PW95]. The concept, termed *spray rendering*, uses smart particles with initial attributes such as colour,
trajectory, life span and an associated behaviour. Depending on the visualization task the user can select different types of particles such as particles for the detection of isosurfaces or the tracking of flow directions.

A general drawback of using particles is the difficulty in perceiving their 3D position in space. Several techniques for improving 3D perception are explained in subsection 4.7.2. Alternatively particles can be replaced by small spheres [MR90]. Spheres require more screen space but improve depth information due to shading and overlay.

4.6.5 Height Fields

Height fields can be used to visualize the scalar field over a surface by constructing an offset surface where the offset at each surface point is proportional to the underlying scalar field. Height values can be emphasized by line segments connecting the height field and the base surface [NFHL91]. The main advantage of height fields is their accurate display of quantitative information.

In general height fields are constructed over a planar domain and the offset is normal to it. Curved base surfaces make it difficult to perceive height values and might result in a self-intersection of the offset surface. Sometimes self-intersection can be avoided by choosing a radial projection from the centre of the domain [NFHL91].

4.6.6 Direct Volume Rendering

Scalar volume data can be directly converted into an image using direct volume rendering (DVR). The technique interprets scalar values as density values which are associated with colour and transparency values using transfer functions.

Two types of direct volume rendering algorithms exist. The ray casting method traces for each image pixel a ray through the volume and accumulates colour and transparency information along the ray [Lev90, SK94, WKB99]. Ray tracing uses additional recursive rays to obtain reflections and refractions [Lev90].

The splatting method is an object-order technique and maps volume data directly into an image by accumulating colour and transparency values there [Wes91]. In general splatting is faster than ray casting but produces lower quality images. Fast splatting methods with improved image quality have been proposed in [MMC99, Kul01]. An example is given in figure 4.5.

Volume images can be generated faster by extracting a 2D plane of data from the 3D Fourier transformed volume and by applying the inverse transform to this slice only [Mal93, TL93]. Although Totsuka and Levoy [TL93] employ frequency domain lighting and depth cueing techniques Moorhead and Zhu [MZ95] report that the resulting projections lack depth information.

It is also possible to use the wavelet domain for volume rendering. Lippert and Gross propose a ray-casting type method which saves computational cost by making use of the self-similarity between the wavelet basis functions when projecting the wavelet transformed volume back into the spatial domain [LG95]. A later paper sug-
4.6 Scalar Icons

Figure 4.5. Direct volume rendering of a human pelvis data set. The images show muscle tissue (left image) and bone material (right image). Skin is represented as a thin semi-transparent layer (used with permission ©Peter Kulka 2001 [Kul01]).

gests wavelet splats for fast volume rendering incorporating progressive refinement [LGK97].

Volume rendering can be sped up by replacing trilinear with bilinear interpolation and by rendering object-aligned slices using hardware accelerated texture mapping [CCF94, BR98]. A recently presented method [RSEB+00] using the multi-texturing capabilities of the GeForce family of graphics cards [nVi] allows interactive frame rates on a PC. Interactive DVR of unstructured data sets is achieved by using the programmable vertex shaders on current generation commodity graphics cards [WMFC02].

Direct volume rendering involves added complexity if compared with alternative scalar field visualization methods. For example, most DVR implementations require volume data sampled over a regular grid. Techniques without this restriction have been proposed but are in general less efficient (e.g., [MHK95, WvTG96]). A second difficulty is that suitable colour and opacity transfer function have to be found in order to extract meaningful images. Several techniques for automatically finding transfer functions for general volumes [KD98, FBT98, FTAT00] and photographic volumes [ERY02] have been proposed. Rheingans and Ebert suggest that finding good transfer functions requires substantial hand tuning and propose instead a non-photorealistic rendering technique to enhance important features [RE01]. Finally image quality is dependent on the reconstruction kernel [ML94, MMK+98] and the normal estimation scheme [MMMY97] employed.

Kim et al. [KWP01] suggest techniques for comparing DVR algorithms and present initial results.
A simplification of DVR is the maximum intensity projection (MIP) which is obtained by computing for each pixel of the view plane the highest data value along a viewing ray through that pixel. Since the method lacks depth information interactive viewing techniques and visual cues are necessary in order to understand the data. The method produces x-ray like images and is frequently used for the visualization of vascular structures in medical data sets (angiography). Several implementations offering interactive speed have been described in the literature [HMS95, MHG00].

Hauser et al. [HMBG00, HMBG01] devise a technique to fuse DVR, MIP, surface rendering and nonphotorealistic rendering. The authors suggest that this approach is especially useful when visualizing inner structures together with semi-transparent outer layers, similar to the focus-and-context approach known from information visualization. An implementation which allows interactive exploration of volume data sets is also presented.

An interesting new techniques is Spectral Volume Rendering proposed by Noordmans et al. [NvS00]. The approach abolishes the association of voxels with RGB colour and opacity values and instead models spectral changes in light caused by interaction with the material within a voxel. The method allows interesting transparency effects and makes it easy to reveal hidden structures in a volume.

Feature rich visualizations can be obtained by replacing one-dimensional transfer functions with multidimensional ones and by applying these to scalar or multivariate data [KKH02]. The authors also suggest a set of manipulation widgets for the intuitive and convenient specification of transfer functions.

A completely different approach is presented by Driver and Buckalew [DB91] who consider a scalar field as a light source and employ a radiosity type technique to display the resulting illumination on the three coordinate planes. The idea suffers from a poor display of 3D structures but has the advantage that regions of interest hidden from view are still visible because of the light cast by them.

4.6.7 Scalar Field Topology and Features

The visualization icons presented so far display a scalar field over a user-defined visualization domain by encoding the field values for each point of the domain. For example, particle fields show scalar field values at selected points, a colour mapped surface encodes field values over a surface and direct volume rendering visualizes a scalar field over a volume.

A different approach is to create visualization icons which reflect the entire structure of a scalar field without representing individual data values. One such icon for 2D scalar fields is the scalar field topology proposed by Bajaj and Pascucci [BP98a]. The topology is defined by connecting critical points (local maxima, minima, and saddle points) by integral curves which are tangent to the scalar field’s gradient field. Additional structure is displayed by inserting contour lines.

A topological analysis of scalar volumes has been proposed by Fujishiro et al. [FTAT00]. The authors employ a 3D generalization of Reeb graphs which encode topological changes between consecutive isosurfaces in the volume.
For time varying scalar fields it is often desirable to observe higher level phenomena. This can be achieved by detecting features and by tracking their evolution over time. Features can be defined as thresholded connected regions and are usually represented as isosurfaces [SSZC94, SW98]. Problems arise due to merging, splitting, creation and death of features. A general feature tracking algorithm is presented in [SW98]. Features can also be used to enhance a visualization by colour coding individual regions [SW97].

### 4.6.8 Visualization Techniques for Multivariate Data

So far this section has introduced visualization icons for individual scalar fields. In some instances scientists are interested in finding relationships between multiple scalar fields. This can be achieved by using techniques from multivariate analysis which is concerned with detecting correlation in multivariate data [CC80].

Multivariate as defined in section 4.3 can be considered as a selection of elements $e_i$ (rows of data), each with $n$ observations $(x_{i1},...,x_{in})$ (variables or columns of data). The challenge in visualizing multivariate data is to maximize the number of displayed variables while showing spatial relationships between different variables without causing mutual interference [RL95]. Visualization methods for multivariate data can be divided into three major groups: geometric projection techniques, layout manipulation techniques and attribute mapping.

Geometric projection techniques are common in multivariate analysis and can be considered as advanced data reduction transformations. Examples include principal component analysis and multidimensional scaling introduced in section 4.3. A further popular method is projection pursuit which attempts to find interesting low-dimensional projections using a so-called pursuit index [CBCH95]. The effectiveness of this and other techniques can be increased by using a Grand Tour which gives a dynamic overview of multiple low-dimensional projections [CBCH95].

Layout manipulation techniques interpret multivariate data as a high-dimensional space with one dimension for each variable (column of the data) and a point for each case (row of the data) [YR91]. Examples are parallel coordinates, world-within-worlds and matrix coordinates.

**Parallel coordinates** represent variable spaces as uniformly spaced parallel vertical coordinate axes [ID90]. Each point in the data set is represented by a polyline intersecting a parallel coordinate axis at the corresponding coordinate (=variable) value of the point. Correlation between variables is perceived as a pattern or common behaviour of the plots. Wegenkittl et al. present several modifications which allow improved perception of structure within a data set [WLG97].

**World-within-world** is a technique in which two or three of the variables are chosen in order to describe a 2D or 3D slice of the multivariate data set, respectively. Previously not chosen variables can be added by embedding a sub-frame into the resulting coordinate frame. The process is either repeated recursively or the sub-frame is used to display data using conventional visualization techniques [FB90]. If only 2D subframes are chosen and data points are represented by individual pixels
the technique is often referred to as *dimensional stacking* [War94].

*MMatrix coordinates* is a technique for studying $k$ variables by arranging them into a matrix with shared scales and by creating graphs comparing each variable with every other variable [Cle85]. If the employed graph is a scatter plot the technique is termed *scatterplot matrix*. The brain’s ability to identify patterns makes it possible to quickly identify relationships between the variables. Interactive computer graphic methods can further improve the displayed information content. An example is *data brushing* in which data points selected in one matrix element are highlighted in all matrix fields which facilitates the discovery of relationships between subsets of the data [WB96, WB97].

In contrast to layout manipulation techniques, attribute mapping interprets multivariate data as defined over a low dimensional space with several variables for each data point. The technique usually involves glyphs which Ribarsky et al. define as graphical representations whose attributes, such as position, size, shape, colour, and orientation, are bound to data [RAEM94]. The authors present a visualization tool which allows the interactive creation of glyphs for multivariate data sets [RAEM94].

A popular example for multivariate glyphs are *Chernoff faces* [Che73, MER99] which encode data by facial features and make use of the brain’s ability to quickly perceive changes in facial expression. The technique can be extended to other familiar shapes with variable components. Chuah and Eick, for example, encode data by the wings, head, tail, and body of an insect shaped glyph [CE98].

Ebert et al. [ERS+99, SHB+99] explore the use of shape in the visualization of multidimensional data by using superquadrics, fractals, and isosurfaces as shape generating entities. In fractal shape generation the base shape is perturbed with a high frequency pattern which does not reduce perception of the base shape. Superquadrics can encode up to two dimensions by varying the two exponents in their mathematical formulation. The parameters control the convexity and the concavity of shapes such that they are visually distinct yet related. Similarly spherical shapes with blobby bulges have been suggested where the vector between the centre of a blob and the centre of a glyph represents the value of a scalar [ERS+99, RSE99].

Variables have also been encoded using star or wheel shaped glyphs where colour, length, and shape of each segment encode information [CE98]. Pixel based techniques map each dimension to a pixel and combine the pixels to a square or rectangle-shaped array. The resulting pixel arrays for each data point are in turn arranged in various fashions to form the final visualization [KK94]. Finally stick icons can be used to encode data by the orientation and length of individual components. If the icons are displayed in a high-density representation the visual effect is that of a texture with differently textured regions indicating structure in the data [EG95].

### 4.7 Vector Icons

Vector fields are a common entity in scientific and engineering data. Examples are displacement fields in elasticity theory and velocity fields in computational fluid dynamics (CFD). In general vector fields have an orientation and are then termed
signed, though unsigned vector fields, such as eigenvector fields, also exist.

Many visualization techniques for vector fields were specifically developed for velocity fields. We therefore use in this section the terms vector field and flow field interchangeably. A couple of definitions are necessary to facilitate the following discussion. A steady flow is a flow field which is constant over time. Time-varying flows are termed unsteady and can be further differentiated into laminar flows and turbulent flows. Laminar flows are characterized by layers of fluid elements with similar velocities whereas in turbulent flows the velocities in neighbouring fluid elements vary randomly.

### 4.7.1 Vector Glyphs

The straightforward way to represent vector fields is by drawing arrows or line segments at individual points with the direction and length of the glyph indicating the length and magnitude of the vector at each point. If the domain is two-dimensional the resulting arrow plots are sometimes called hedgehogs [Pv94].

Drawbacks of this representation include the visual system’s attempt to interpolate line segments [KH91]. Also, since vector glyphs cover multiple pixels on the screen, it is not intuitively clear which value a glyph represents, i.e., is the location of the visualized vector value the centre, start or the end point of the glyph?

One solution is to indicate the location of the visualized vector value by a point or a sphere or by marking the location on the arrow glyph using a stripe. Rather than using a vector arrow it is also possible to use a short streamline (subsection 4.7.3) with a length proportional to the vector magnitude [WFL+00]. Other alternative icons are presented by Klassen and Harrington [KH91].

Care has to be taken that the grid distance of an arrow plot is smaller than the maximum vector length since otherwise visual cluttering due to the intersection of glyphs representing different data points may occur. Perception of vector plots can be improved by animating line segments using colour cycling [YM95].

The above mentioned techniques can also be employed over 3D domains. However, using 2D icons in three dimensions makes it difficult to perceive the 3D position and orientation of the represented vectors. A solution is to include shadows of the vector icons when visualizing planar sections of a 3D vector field [KH91]. Alternatively the out-of-plane component can be visualized using colour mapping [WMK+99]. Max et al. [MCG94] visualize vector fields near contour surfaces by growing little hairs out of the surface and have them bend with the flow field.

Improved perception of a 3D flow field is obtained by using three dimensional glyphs and by illuminating the icons. For example a 3D arrow can be composed of a cylinder and a cone. Haswell [Has95] proposes a hollow cone with different surfaces on the inside and the outside which not only reinforces vector direction but also allows immediate perception of the vector orientation. In general 3D glyphs require more screen space then the 2D equivalent and therefore reduce the amount of represented data. Furthermore illumination can compromise the perception of colour if information is encoded by colour mapping an icon.
The main difficulty with using vector glyphs is to distribute them optimally over the domain such that perception and encoded information is maximized. The problem magnifies if the underlying vector field has large variations of vector magnitude. Furthermore, since vector glyphs are not spherically symmetric the optimal sampling density varies vastly in each dimension [MZ95]. Flow features such as vortices are especially difficult to identify.

A more uniform visual display can be achieved by normalising the glyphs and representing the vector magnitude by colour information or other attributes. Dovey presents a technique to distribute vector icons uniformly over arbitrary surfaces [Dov95]. A completely different approach for representing vector fields with glyphs is given by employing vector field simplification as a data reduction scheme and using one glyph only for each resulting cluster [Tv99, HWHJ99]. The resulting representations are sparse yet information rich.

Figure 4.6. Vector glyph representing the local field behaviour in a fluid flow (© 1994 Academic Press [Pew94]).

The vector glyphs discussed so far represent the field information at only a single point. In many instances it is useful to also know the local field behaviour in the neighbourhood of that point. Neighbourhood information is encoded by a flow field probe devised by de Leeuw and van Wijk [dv93, vHdP94]. The glyph represents properties derived from the local velocity gradient tensor and is illustrated in figure 4.6. The length, curvature and the candy stripes of the cylindrical shaft visualize magnitude, local streamline curvature and rotation of the flow field, respectively. A half ellipsoid at the bottom of the shaft encodes acceleration of velocities and convergence or divergence is described by bending the circular membrane so that it is
everywhere orthogonal to the flow field. Finally shear is encoded by the angle of a ring shaped surface with respect to a reference frame.

Alternative glyphs can be constructed for topological flow features (special points in the vicinity of which the flow direction changes suddenly) and are explained in subsection 4.7.8.

4.7.2 Particle Advection

A simple method to visualize vector fields is to distribute a set of particles over the domain and to advect them with the vector field. Flow direction and speed can be emphasized by blurring the particles in the direction they move. The method is particularly effective for the visualization of fluid flows since the particle representation is perhaps the most realistic one [Pv94] and is therefore intuitive and easily understood. Particles are especially well suited for turbulent flows where icons computed by integral curves and surfaces (see next section) become highly irregular [vHdP94].

Potential drawbacks are the lack of interactivity if the particle number is too high and difficulties in perceiving the 3D structure of the flow. Also the animation of particles by vector field advection might lead to the evolution of uneven particle distributions [WG97]. This disadvantage can not occur when animating textures and colours [WG97] (see subsection 4.7.5). A similar idea has been employed by van Gelder and Wilhelms who animate particles using hardware supported colour interpolation and discrete colour mapping [vW92].

The 3D perception of particle systems has been improved by Stolk and van Wijk [Sv92, van93a] who introduce surface particles which are modeled as motion blurred points with a surface normal. Distortions due to speed and flow direction are revealed by employing spherical particle sources with discrete release time. An example is given by the red and yellow ellipsoidal shapes in figure 4.7. The texture generated by blurred particles helps to discriminate between overlapping surfaces. For low density particle clouds good images are achieved without considering occlusion whereas for high density particle surfaces depth sorting and z-buffering improves the perception of the 3D geometry [van93a].

Max et al. modify this technique and achieve interactive speed by taking advantage of hardware rendering [MCG94]. The authors generate particles only in the vicinity of contour surfaces. Since this restriction reduces depth ambiguities surface normals for particle shading are dispensable. Particles leaving the contour vicinity are deleted and new particles are created such that a constant particle density is maintained.

An efficient visualization of turbulent flow is achieved by modelling particle motion as a smooth convective motion with a random perturbation [HP93, vHdP94]. The method is based on Reynolds-average equation which is used in the simulation of turbulent flows and gives an impression of the distribution of local flow dynamics without reconstructing the complete turbulent motion of individual particles.
Figure 4.7. Surface particles visualizing a turbulent flow (©1994 Academic Press [Pv94]).

4.7.3 Integral Curves and Surfaces

The previously introduced icons represent vector data only at discrete points. A continuous representation of particle paths in a steady flow is obtained by using streamlines which are everywhere tangential to the underlying vector field. Mathematically a streamline can therefore be described as an integral curve \( x(s) \) which satisfies

\[
\frac{dx}{ds} = v(x(s)) , \quad x(0) = x_0
\]  

(4.2)

where \( v(x) \) is a vector field and the initial condition \( x(0) \) defines the starting point \( x_0 \) of the streamline.

In general the above system of equations has no analytic solution and is solved by numerical integration. Standard techniques for streamline integration include fixed step size integrators such as the Euler, Midpoint or Runge-Kutta method. A faster computation can be achieved by adaptive step size integration [PVTF92, HWN93]. If the step size is too large or the curvature is too high a dense sampling of the streamline might be required in order to obtain a good visual approximation of it. The sampling can be performed as a post-integration interpolation step [SH95] or by using a specialised integrator which produces an interpolation from the integration information [HWN93, pp.176].

Note that a streamline for an unsigned vector field, such as an eigenvector field, must be integrated in both the positive and the negative direction of the unsigned vector field.
The general advantage of streamlines is their intuitive representation and that they reduce visual cluttering by replacing multiple vector glyphs, such as vector arrows, by a single line. Additional information, such as vector magnitude, can be encoded by colour mapping the streamline.

One problem with streamlines is the difficulty in placing them. Even in 2D accurate control of streamline density is the key to producing effective visualizations of vector fields. If streamlines are too close together they merge and the directional information is obscured whereas if the streamlines are too far apart information is missing for large areas.

Several authors have explored methods for finding a uniform distribution of streamlines. Turk and Banks minimise an energy function in order to place streamlines at a specified density. The energy function expresses the difference between a low-pass filtered version of the current image and the required visual density [TB96]. Mao et al. extend the technique to distribute streamlines over a 3D parametric surface as found in curvilinear grids [MHHI98]. Jobard and Lefer [JL97a] present a faster alternative method which is an extension of earlier work from Max et al. [MCG94]. The algorithm grows streamlines from seed points which are a minimum distance $d_{sep}$ apart from all existing streamlines. A streamline grows until it leaves the domain, ends at a sink or comes closer than $d_{test}$ to another streamlines. The authors recommend a value $d_{test} = 0.5d_{sep}$ in order to obtain an aesthetically pleasing image with long streamlines. A related method proposed by Wegenkittl and Gröller [WG97] places short streamlines (streamlets) into a 2D image by using a distance image with the same resolution as the output image. Each pixel of the distance image contains the distance to the closest streamlet. If a streamlet can be successfully placed the distance image is updated.

The problem of placing streamlines becomes even more difficult in 3D. A 3D extension of the streamline placement algorithm by Jobard and Lefer [JL97a] is found in [FG98]. Zoeckler et al. generate seed points for streamline integration by dividing a volume into uniform cells [ZSH96]. Each cell is given a degree of interest depending on vector magnitude and other scalar entities inside the cell. Random seed points are then chosen within cells which themselves are selected randomly with a probability proportional to their degree of interest. Stalling et al. [SZH97] suggest several techniques to interactively select seed points for streamline generation. Seed volumes provide seed points randomly distributed over a volume with a density depending on a degree-of-interest function usually given by another scalar field over the domain. Similarly seed surfaces can be constructed by triangulating a modelled or a derived surface (isosurface) and seed lines can be interactively constructed by intersecting surfaces, such as isosurfaces, with cutting planes.

Once an effective set of streamlines for visualizing a 3D vector field is found the problem remains that the resulting scene is often complex and hard to interpret due to the lack of depth information. An improved perception of 3D geometry is achieved by replacing streamlines with illuminated cylindrical tubes. Since each such defined streamtube is composed of multiple polygons the frame rate and spatial resolution of the visualization is reduced.
A better solution is illuminated field lines which were proposed by Stalling et al. [SZH97, ZSH96]. Illuminated field lines are semi-transparent streamlines which are rendered using hardware supported texture mapping and a Phong-like illumination model. A dense representation of a vector field is achieved by distributing the streamlines pseudo-randomly over a region of interest. Illuminated field lines can be animated by changing texture coordinates or other attributes [Cur01].

Using the concept of streamlines more advanced visualization methods can be derived. The following paragraphs introduce several two or three dimensional constructs formed by connecting multiple streamlines. The resulting icons can be illuminated and have local parameters mapped on them [BHR+94]. Advantages of such representations are that display ambiguities due to the lack of shading information are resolved and that additional information about local flow behaviour in the vicinity of the trajectory can be encoded by the shape of the construct.

An example is stream ribbons which were proposed by Volpe [Vol89] as a method to show translation, angular rotation, and shear deformation in a flow field. The original implementation uses two streamlines and connects them by a mesh of polygons. Ueng and Ma construct a stream ribbon from a streamline and a constant length normal vector at each point of the streamline. The normal vector is rotated around the streamline using the rotation (curl) of the vector field [USM95, USM96]. Interesting see-through effects have been achieved by rendering stream ribbons with surface particles [van93a].

A generalisation of stream ribbons are stream surfaces which are defined by adjacent streamlines originating from a line (rake) or a curve perpendicular to the flow [Hul90]. A basic implementation of this technique traces only a couple of streamlines and connects them by polygons [Hul92]. Problems due to convergence, divergence and shear flow have been partially solved by Hultquist who uses a particle tracing method. The author inserts particles into the flow front if divergence becomes too strong and he deletes particles if the convergence exceeds a given limit [Hul92].

For incompressible flows a stream surface can be defined implicitly as the solution of the equation \( f(x) = c \), satisfying the condition \( \nabla f \cdot \mathbf{v} = 0 \). The unknown function \( f \) is computed iteratively for all grid points by solving the convection equation for incompressible flows. The solution is based on the idea that the streamsurfaces are isosurfaces in the concentration field resulting from simulating the convection process for a given concentration of ink at the inflow boundaries.

An algorithm for extracting implicit stream surfaces at interactive frame rates is proposed in [van93b, vHdP94]. Cai and Heng present an extension which automatically generates implicit stream surfaces that properly depict the topology of an irrotational fluid flow [CH97]. Problems occur if the streamlines cross singularities and saddle point type critical points. Stalling et al. present an algorithm which considers these difficulties [ZIBb] and computes stream surfaces for general vector fields.

Löffelmann et al. suggest several strategies to improve visual perception of stream surfaces in fluid flows [LMGP97]. Inserting semi-transparent arrow shaped “windows” into the surface reduces occlusion of other scene constructs. Additionally the
4.7 Vector Icons

distorted shape of the arrows indicates velocity and divergence of the flow. Perception of the 3D geometry is improved by employing cutting planes to divide the stream surface into portions rendered using different attributes. Flow within stream surfaces is emphasized using an anisotropic spot noise texture. The authors employ hash-shaped spots and align the texture space with the parameter lines of the stream surface which corresponds to streamlines and time lines [LMGP97]. In contrast to Line Integral Convolution (LIC) textures (see subsection 4.7.6) not only the flow direction within the surface is visualized but also flow fronts.

The principles used in the construction of stream surfaces can be extended to 3D to create flow volumes. Instead of a linear particle source a polygonal particle source is chosen and the resulting streamlines are connected to triangular prisms by using a curvature-based adaptive subdivision [MBC93, CMB94]. If the flow diverges new streamlines are inserted and a finer subdivision is generated. The resulting volume is subdivided into tetrahedra and displayed as semi-transparent smoke or clouds using tetrahedral volume rendering. Clouds can be animated by varying the texture coordinates. Varying the opacity of the smoke with each time step can generate interesting effects such as smoke puffs. An efficient calculation of flow volumes for unsteady flows is presented in [BLM95].

A streamtube is defined as the set of streamlines that originate from a circular curve perpendicular to the flow. A streamtube can be efficiently constructed by sweeping a circular cross section along the streamline with a radius dependent on the cross flow divergence [USM95, USM96]. An alternative definition for a streamtube sweeps a polygon along a streamline and deforms the polygon by the local strain and rotation. The radius and the sides of the polygon can be used to encode additional scalar fields such as the velocity, vorticity and divergence [SVL91].

A range of alternative methods originating from CFD experiments have been proposed to visualize time dependent vector fields: Streaklines are generated by continuously injecting particles, e.g., dye, into a flow. For a computer simulation a streakline at time $t_0$ is visualized by joining the positions of all particles released at previous time steps. As with streamlines, streaklines can be extended to 3D to form flow volumes. Pathlines describe the path of a single particle released into the flow and are therefore represented by a polyline connecting the current particle position with its position at all previous time steps. Note that for steady flows streaklines and pathlines are identical to streamlines. Finally timelines are rakes of connected points released simultaneously from a linear source and time surfaces are defined as grids of connected points released from a planar source [vHdP94]. Efficient methods for computing streaklines, pathlines and timelines in time dependent flows with a moving curvilinear grid have been proposed by Kenwright and Lane [KL95, KL96].

Many computational problems with flow surfaces and flow volumes can be avoided by the streamball technique proposed by Brill et al. [BHR+94]. The method uses particles to visualize the flow but models each particle as the source of a potential field. An individual particle is then rendered by an isosurface of its potential field. If particles are sufficiently close together the potential fields overlap and particle isosurfaces fuse together to form tube and surface-like structures. An extension of
the method uses continuous field lines, such as streamlines, as skeletons for potential fields in order to obtain smoother looking flow representations. The method eliminates numerical problems such as when to split and merge stream surfaces and can display additional information such as the velocity of a flow field by the distance of individual streamballs.

Wegenkittl et al. [WLG97] have generalised streamlines to visualize higher dimensional dynamical systems described by a system of differential equations. The resulting trajectories with wings are constructed by using a base trajectory with several offset curves added for additional dimensions.

Related to the concept of integral curves and surfaces is the technique of vector warping [SML96a]. The underlying idea is that vector data is often associated with motion in the form of velocity or displacement. A vector field can therefore be visualized by applying it to an object and depicting the resulting deformed structure.

### 4.7.4 Multivariate Colour Maps

Subsection 4.6.1 introduced colour mapping for the visualization of scalar fields and presented bivariate colour maps as a method to encode two independent scalar variables. The same principles can be used to encode the two independent variables of a 2D vector field (e.g., x- and y-coordinate or angle and magnitude). In contrast to the vector icons introduced so far colour mapping generates a continuous vector representation over a domain. In addition colour is invariant under projection, i.e., the bivariate colour map always displays the same information whereas a vector icon might give misleading information when viewed from different angles. A drawback of colour mapping is the poor representation of quantitative information. As a result perception of the displayed vector information, usually orientation and magnitude, is limited. Also the representation is non-intuitive so that perceived information might be limited for an inexperienced user.

An example of bivariate colour mapping is the colour wheel techniques by Johannsen and Moorhead [JI94, JI95]. The authors employ the HSV colour model and represent vector direction by the hue and vector magnitude by both saturation and lightness. A log scale mapping for the vector magnitude reflects the visual system’s nonlinear perception [MZ95]. The representation shows zero vectors in black and indicates vortices in the vector field by rainbow coloured whirls.

Similarly Boring and Pang suggest mapping the vector direction to the HSV value and the vector magnitude to the HSV hue [BP96b]. The authors represent the vector direction by the angle with the light source so that, as with bump mapping, vectors pointing away from the light source appear dimmer. Alternatively roles can be reversed, i.e., the vector direction is mapped to hue which results in a higher differentiation of angular differences.

Kindlmann and Weinstein use colour maps to visualize 3D vector fields. The authors encode the direction of a normalised 3D vector by using a spherical colour map which indicates the polar coordinates of the vector [KW99]. Moorhead and Zhu [MZ95] summarize a technique by Hall [Hal93] who visualizes vector fields using
4.7 Vector Icons

perceptually-based colour spaces and direct volume rendering. According to the authors Hall uses a perceptually uniform color space like the Munsell space in order to encode the complete 3D vector information by the 3D colour space. Moorhead and Zhu suggest that this idea is reaching too far and that limiting the paradigm to a 2D space may be wise. Furthermore the authors note that global variations can be emphasized by quantizing the color-space in order to produce abrupt edges.

4.7.5 Texture-based Methods

All vector icons introduced so far have a limited spatial resolution with the exception of bivariate colour mapping which, however, suffers from a weak and unintuitive encoding of directional information. Both of these drawbacks can be avoided by using textures which are able to clearly encode orientational information over both continuous 2D and 3D domain. The technique also avoids the problem of visual cluttering (common to streamlines and vector glyphs) and relieves the user of the task of placing icons in order to find interesting features.

One of the first applications of textures in scientific visualization has been presented by Upson et al. who use large clouds of semi-transparent advected particles to create smoke or cloud like textures in order to visualize velocity fields [UFK+89].

Ware and Knight [WK95] encode scalar and vector data using the size, contrast and orientation of a texture. The texture is produced using Gabor functions which model the visual system’s processing of spatial information by spatial frequency analysis. The technique suffers from its high computational complexity and the fact that the texture is sine based. An example is given in figure 4.8.

Spot noise is used to synthesize textures over curved surfaces in order to visualize vector fields. The method distributes a large number of spots over the surface and transforms them according to the underlying vector field. The original algorithm [van91] stretches spots elliptically along the vector field which results in a weak representation of small scale structures. An improved version by de Leeuw and van Wijk bends spots along a local stream surface [dv95, dPPW95, LLN]. Since the total spot area is kept constant an increased vector magnitude is encoded by a finer texture granularity. The authors refine texture details by using high-pass filtered spots. The representation can be animated by advecting the spots.

Similar results as with spot noise are produced with Line Integral Convolution (LIC) which is explained in subsection 4.7.6. A comparison of spot noise and LIC is given in [dv98]. Related to both of these techniques is a method of Preußer and Rumpf [PR99b] who use nonlinear anisotropic diffusion to smooth an input noise texture along streamlines while sharpening it in the orthogonal direction. The authors also discuss methods to extend the algorithm to 3D by using properties of the diffusion texture to open up the view in inner regions.

Jobard and Lefer [JL97a] generate a texture by placing streamlines close together until the visualization domain is covered. The authors then separate streamlines by mapping a periodic intensity function onto the streamlines. Aliasing effects due to intensity differences between adjacent streamlines are removed by a blur filter.
Figure 4.8. Magnetic field visualized using a texture generated by Gabor filters. Field orientation, strength, and potential are mapped to texture orientation, contrast and inverse size, and colour, respectively (©1995 Association for Computing Machinery [WK95]).

A similar method can be employed for animating steady flows and involves the creation of a dense streamline representation, the computation of a velocity map and a colour table animation [JL97b]. In later papers Jobard, Erlbacher and Hussaini advect textures directly by using hardware acceleration [JEH00] and a Lagrangian-Eulerian approach [JEH01, JEH02]. In both cases multiple frames of the texture are blended together in order to introduce spatial correlation into the images. Adding random noise to the texture avoids loss of detail in divergent regions.

Verma et al. [VKP99] present a related technique called Pseudo Line Integral Convolution (PLIC). Images spanning the spectrum from LIC to streamline representations are generated by varying a single parameter.

Löffelmann et al. propose the use of virtual ink droplets to visualize vector fields over a plane [LKG97]. Ink droplets are modeled as height fields and are distributed randomly over the surface. The ink is then propagated over the surface according to the underlying vector field while the height is reduced according to the amount of ink absorbed by the underlying “paper”. The method visualizes the orientation of a vector field similar to oriented LIC (subsection 4.7.6) but according to the authors is up to 200 times faster. The method suffers from inaccuracies due to the merging of ink droplets and no comparison with FROLIC (fast rendering of oriented LIC) is given.

The techniques described above all define textures using mathematical principles.
An alternative methodology associates vector information with brush strokes in an oil-painting. Kirby et al. [KML99] utilize techniques from oil painting to combine multiple layers of visual elements to a surface texture encoding multiple fields. One application is to display simultaneously flow direction and related data such as velocity, vorticity, divergence, strain rate and shear.

### 4.7.6 Line Integral Convolution

Line Integral Convolution (LIC) is an effective method to visualize vector fields by using curvilinear filters to locally blur an input noise texture \( I \) along a vector field \( \mathbf{v} \). The algorithm, as originally proposed by Cabral and Leedom [CL93], assumes square pixels for the input and output texture map and its steps are indicated in figure 4.9.

\[ p_i = p_{i-1} + \frac{\mathbf{v}(p_{i-1})}{\|\mathbf{v}(p_{i-1})\|} \Delta s_{i-1} \]

and \( \Delta s_{i-1} \) is the distance to the pixel boundary, \( s_0 = 0, s_{i+1} = s_i + \Delta s_i \), and \( i = 0, \ldots, l \) where \( l \) is chosen such that \( s_l \leq L < s_{l+1} \).

Pixels intersected in the backward direction are computed analogously and are indicated by negative indices \( i = -\hat{l}, \ldots, 0 \) where \( \hat{l} \) is chosen such that \( s_{-\hat{l}} \leq L < s_{-(\hat{l}+1)} \). For each line segment \( [s_i, s_{i+1}] \) of the streamline intersecting pixel \( p_i \) an exact integral of a convolution kernel \( k(w) \) is computed and used as weight in the LIC

\[ h_i = \int_{s_i}^{s_{i+\Delta s_i}} k(w) dw \]
The output pixel $O(q,r)$ is then given by

$$O(q,r) = \frac{\sum_{i=-\hat{l}}^{\hat{l}} I(p_{i,x}, p_{i,y}) h_i}{\sum_{i=-\hat{l}}^{\hat{l}} h_i}$$

In the simplest case the convolution kernel is a box filter so that the output texture represents the weighted input texture along the streamline. Vector magnitude is represented either by using colour mapping or by varying the length $L$ of the filter kernel.

Parameters influencing the quality of the output texture are the input texture, the filter kernel, and the length $L$. Most authors employ an input texture based on white noise which has a constant power spectrum and is completely random. Aliasing effects due to high frequency components in the white noise texture can be reduced by low-pass filtering the input texture [CL93]. Verma et al. demonstrate that any correlation in the input texture is increased by the vector field which leads to artifacts [VKP99]. Okada and Lane obtain good output images with high contrast by applying LIC twice, i.e., the authors use the LIC image as an input texture for a second application of the algorithm [OL96]. Kiu and Banks [KB96] present multi-frequency noise for LIC. Regions of low and high frequency noise in the input texture appear as long fat and narrow short streaks in the output texture and can be used to symbolise high and low velocities, respectively. The method can be extended to a curvilinear grid by generating the noise texture using Poisson ellipse sampling [MHK+98].

The choice of a filter kernel depends on the intended application. For simple images a box filter is sufficient so that the output texture represents the weighted input texture along the streamline. For animations a varying phase shifted periodic filter kernel can be used [CL93]. Wegenkittl et al. [WGP97] propose an oriented LIC (OLIC) which convolves a sparse texture with a stump like kernel. The resulting image resembles ink droplets smeared along the vector field direction. Fast rendering is achieved by approximating the “ink droplet traces” by a series of overlapping disks. The algorithm (called FROLIC) is related to particle based methods and produces only an approximation to LIC [WG97].

Recommended values for the filter length $L$ range from 10 times the pixel size [CL93] to $\frac{1}{20}$ of the output image size [SH95]. A large value of $L$ reduces contrast in the output texture whereas too small a value of $L$ gives insufficient filtering. The first effect can be decreased by amplifying the input image or by contrast stretching the output image. Several example of LIC using a box filter with varying length are shown in figure 4.10.

Stalling and Hege [SH95] suggest a modification of LIC called Fast LIC (FLIC) which is an order of magnitude faster, more accurate and resolution independent. The authors recognise that for a constant filter kernel (box filter) the convolution integral of two consecutive pixels on a streamline differs only for a couple of pixels. To formalise the difference it is convenient to reparameterize a streamline $\sigma(u)$ in
the vector field \( \mathbf{v} \) with its arc-length \( s \)

\[
\frac{d}{ds} \sigma(s) = \frac{d\sigma}{du} \frac{du}{ds} = \mathbf{v}(\sigma(s))
\]

A streamline through the point \( p_0 \) is then defined by solving the above ordinary differential equation with the initial condition \( \sigma(0) = p_0 \). Assume two points \( p_1 = \sigma(s_1) \) and \( p_2 = \sigma(s_2) \) are only a small distance \( \Delta s = s_2 - s_1 \) apart then the output texture can be computed as

\[
O(p_2) = O(p_1) - k \int_{s_1-L}^{s_1-L+\Delta s} I(\sigma(s))ds + k \int_{s_1+L}^{s_1+L+\Delta s} I(\sigma(s))ds
\]

By calculating long streamlines and only two correction terms for each new pixel redundancies present in the conventional LIC algorithm are avoided.

For the implementation of FLIC the convolution integral is computed by sampling the input texture at evenly spaced locations. If the number of sample points hitting an input pixel is smaller than some minimum number a new streamline is started. The output texture is then computed by normalizing each pixel with its

---

**Figure 4.10.** Top row: LIC with a kernel length of 40. From left to right: using white noise, using low pass filtered white noise, using low pass filtered white noise and contrast stretching the output texture. Bottom row: kernel length of 10, 20, and 160. All images are contrast stretched and use low-pass filtered white noise.
number of hits. According to the authors a step size of $h = 0.5$ times the width of a texture cell is sufficient. If the output image has a higher resolution than the input texture than the step size has to be decreased correspondingly.

In contrast to the original LIC algorithm FLIC avoids aliasing due to the correlation between pixel intensities along a streamline. In addition the technique allows the use of higher order numerical integration methods whereas the traditional LIC technique essentially employs an Euler method. Because of the discontinuities along cell boundaries the authors found traditional Runge-Kutta methods with a limited maximum step size to be sufficient. The integration steps can become so large that cubic Hermite interpolation is necessary for texture sampling.

A drawback of FLIC is that only simple convolution kernels can be employed. An extension using piecewise polynomial filter kernels is presented in [HS97].

Several other extensions to LIC have been presented in the literature. Forssell applies the algorithm to vector fields over parametric surfaces by computing the LIC texture in material space and by texture mapping it onto the surface in world coordinates [For94]. Mao et al. [MHK*98] suggest multi-granularity noise to reduce distortions in the mapping process. A further extension for general, possibly multiply connected surfaces has been presented by Battke et al. [BSH97]. Three-dimensional vector fields over 2D surfaces have been visualized by varying the saturation and lightness of a LIC texture according to the angle between the surface and the vector [OL96]. Flow features are characterized by different hues. Alternatively a height field can be build from the vector’s normal components [SBH99].

Additional modifications exist for unsteady fluid flows. Forssell and Cohen use streaklines rather than streamlines for the convolution kernel [FC95]. Shen and Kao [SK97b, SK98] achieve improved spatial coherence and more accurate time stepping by using Stalling and Hege’s method [SH95] and by integrating along pathlines rather than streamlines. Yang et al. [YKF00] report that the method is not applicable for MR blood flow imaging due to the limited temporal resolution of the raw data. Instead the authors employ a noise reduction preprocessing step based on the principle of mass conservation in order to obtain a more consistent flow data. Yang et al. represent the flow by a 3D texture with linearly enhanced streamlines by iterating a LIC algorithm and by applying a Laplacian high pass filter at each step [YKF00]. Sundquist computes a FLIC texture by using a “motion” vector field to evolve an FLIC texture for each time-frame via a set of advected randomly placed particles [Sun03].

LIC can be extended to 3D and displayed using direct volume rendering [CL93]. Rezk-Salama et al. present an implementation for interactive exploration of volume LIC using texture mapping [RSHTE99]. Interrante and Grosch improve 3D perception of volume rendered 3D LIC by enclosing streamlines with visibility-impeding halos which indicate depth discontinuities [IG97, IG98]. Shen et al. [SJM96] combine dye advection with three-dimensional LIC in order to visualize local and global flow features simultaneously.
4.7 Vector Icons

4.7.7 Direct Volume Rendering

As for scalar fields, volume rendering is an attractive proposition for vector fields because of its promise to show the entire field all at once. An example mentioned in the previous subsection is direct volume rendering of three-dimensional LIC textures.

Crawfis and Max [CM92, MCW93] represent vectors as antialiased line segments and compose them with a splatting-like algorithm. Opacity and colour of the line segments are used to encode vector magnitude and its normal component. In an alternative implementation Max et al. splat antialiased line bundles oriented along the vector field with colour and position of each line randomly jittered in order to produce an anisotropic texture [MCG94]. Rather than splatting individual lines or line bundles Crawfis et al. [CM93, CMB94, Cra95] suggest splatting an anisotropic texture, e.g., a texture of particle traces, and to rotate the texture to line up with the underlying vector field. The technique is termed *texture splats* and can be interpreted as 3D splatted spot noise.

Frühauf introduces ray casting of volumetric vector data [Frü96]. Vector direction is encoded by lightness and hue using an angle measure dependent on view direction and light direction. Vector magnitude is represented by opacity.

Instead of volume rendering an entire vector field Hong et al. suggest volume rendering a scalar field with vector information only displayed in relevant regions using pre-voxelized vector icons [HMK95]. The technique uses an incremental image update to achieve interactive frame rates and supports interactive viewing tools and real-time animation of vector icons.

Turbulent flows can be visualized by evolving clouds and smoke [EP90]. Subsection 4.7.3 described several techniques to volume render flow volumes as smoke or clouds. King et al. animate such effects in real-time by using graphics hardware and texture cycling [KCR99]. Van Wijk et al. observe that this class of methods represents turbulent motion as fluctuating densities. As a result small-scale turbulent effects which cause only minor spatial and temporal density changes are difficult to perceive [vHdP94].

4.7.8 Vector Field Topology and Features

A vector field \( \mathbf{v}(\mathbf{x}) \) can be characterized by considering its *critical points* which are points with zero vector magnitude. Critical points are the only points where streamlines are non-parallel and therefore indicate important flow features. Furthermore these points can be used for the comparison of vector fields [LBH98, BH99]. A critical point \( \mathbf{x}_0 \) can be classified by considering the eigenvalues of the Jacobian

\[
\mathbf{J}_\mathbf{v}(\mathbf{x}_0) = \left( \frac{\partial v_i}{\partial x_j} \right)_{\mathbf{x}_0} i, j = 1, \ldots, n
\]

where \( n \) is the dimension. The type of a critical point indicates the flow pattern in its immediate neighbourhood.

So far research in this area has concentrated on characterizing critical points of 2D and 3D vector fields. In two dimensions the Jacobian of a vector field is a
$2 \times 2$ matrix and therefore has two eigenvalues with real components $R_1$ and $R_2$ and imaginary components $I_1$ and $I_2$. The type of a critical point and hence the local flow topology depends on the signs of these components. Real components greater or smaller than zero represent repelling or attracting flow features, respectively, whereas non-zero imaginary components symbolise circular flows. The resulting types of critical points are depicted in figure 4.11. Note that these are all types of critical point which can occur in a two-dimensional bilinearly interpolated vector data set. By using a non-linear data representations higher order singularities can be extracted [SHK'+97, SKMR98].

**Figure 4.11.** Classification and example icons for critical points. $R_1, R_2$ and $I_1, I_2$ denote the real and imaginary parts of the eigenvalues of the Jacobian of the vector field, respectively (after [HH89]).

In 2D critical points and boundary points with zero velocity (*attachment nodes* and *detachment nodes*) can be connected by special streamlines (*separation lines*) in order to divide a flow field into regions of similar flow behaviour [HH89, HH90, GLL91]. The resulting image represents the topology of the vector field. An example is given in figure 4.12. Implementation difficulties arise from numerical problems and the fact that separation lines can exist even if the vector field does not contain critical points (*open separation lines*). Kenwright et al. present mathematical methods for the detection of both *open* and *closed separation lines* [Ken98, KHL99].

In three dimensions critical points are classified by recognising that the Jacobian
always has at least one real eigenvalue. The eigenvectors belonging to the remaining
two eigenvalues span a plane for which the flow topology is determined analogously
to the 2D case. The first eigenvector is non-coplanar to this plane and represents
an attracting or repelling flow in the third dimension. In order to compute the 3D
topology it is necessary to detect separation and attachment lines [Ken98, KHL99]
and to connect them to the critical points via streamsurfaces [HH91].

Several interesting observations with regard to vector field topologies have been
made. Scheuermann et al. note that for sampled vector data the computed vector
topology is dependent on the interpolation scheme used [STH99]. De Leeuw and
van Liere [dv99] show that turbulent flows can result in a proliferation of critical
points and a cluttered topological representation. A simplified display is achieved by
collapsing topologies. Similarly multi-level topologies can be obtained [dv00]. The
time evolution of 2D flows can be represented as a third spatial dimension resulting
in topological surfaces as illustrated in figure 4.13.

Figure 4.12. Topology and critical point
glyphs of the velocity vector field of a 3D flow
on the surface of a hemisphere cylinder (©1991
IEEE Computer Society Press [HH91]).

Figure 4.13. A topological surface visu-
alizing the time history of a 2D flow past
a circular cylinder (©1991 IEEE Com-
puter Society Press [HH91]).

So far this section has explained the computation of critical points and their
function in the computation of the vector field topology. Additional information
can be visualized by displaying the critical points themselves using critical point
glyphs. An example by Helman and Hesselink [HH91] is shown in figure 4.12. The
glyphs constitute a 3D extension of the schematic representation in figure 4.11. The
arrows point in the direction of the real eigenvectors with their lengths proportional
to the corresponding eigenvalues. The red or blue colour of the arrow head indi-
cates a negative or positive eigenvalue, respectively. If two eigenvalues are complex
they are represented by disks which are in the plane spanned by the corresponding
eigenvectors. The disk diameters are proportional to the real and imaginary parts
of the scaled eigenvalues. A dark blue or a yellow colour represent a positive or a
negative real part, respectively, and a light blue or a red colour represent a positive
or a negative imaginary part, respectively.

Löffelmann et al. observe that topological analysis gives qualitative information but lacks quantitative information such as the spatial extent of trajectories [Löf98, LDG98]. The authors introduce therefore a special class of vector glyphs which combines both topological and quantitative information. The glyphs are positioned at critical points and indicate the local topology by classifying the local flow pattern as described above. Quantitative information is added by inserting a bunch of streamlets (short streamlines) originating in the vicinity of the critical point at points stochastically chosen depending on the critical point type. The length, direction and colour of the streamlets indicates the velocity, direction and orientation of the flow, respectively.

It is important to note that critical points are not the only topological flow features and therefore are not sufficient for understanding a complex vector field. Additional flow features have been described and characterized mathematically by Asimov [Asi93]. Peikert and Roth present a method to compute global line type features in vector fields [PR99a]. Tang and Medioni present a technique to extract local minima and maxima from a vector field [TM98]. The resulting features are termed extremal surfaces and extremal curves and can be used to extract extreme features such as shock waves in CFD data. Features can also be obtained by using concepts from computer vision and mathematical morphology [SZ93]. Flow features are not only used in visualizations but are also employed for data reduction, localized measurements, and data comparison [SiI95]. This means feature extraction can be an important part of the data transformation stage in the visualization pipeline (see section 4.3).

An important class of features in CFD data sets are vortices since they represent regions of energy loss [PP00]. Vortices are in general characterized by a turbulent swirling flow, though no agreement for a formal definition seems to exist [RP96]. One way to identify vortices is by detecting vortex cores, e.g., by integrating in the direction of the only real eigenvector of a 3D spiral-saddle critical point [GLL91]. Performance can be improved by correlating vorticity data with co-existing physical fields such as pressure [MZ95]. Roth and Peikert show, however, that vortex cores can exist regardless of the presence of critical points [RP96]. Alternative methods for vortex identification are described in [BS94, KH97, PR99a]. Bent vortices have been identified using higher order derivatives [RP98]. Visual representations for vortices include particle systems, streamlines, streamtubes and the placement of a series of colour mapped polygons orthogonal to the vortex axis [KH97]. The associated vortex tubes can be represented using direct volume rendering [MZ94] and twisted tube-like surfaces [BS94].

The visualization of vortices and other flow features in time-varying vector fields poses additional difficulties because features can merge (amalgamate), split (bifurcate) and change type. An overview of these problems is given in [MZ95]. A feature tracking algorithm for turbulent 3D features is presented in [SW97]. Polthier and Preuß suggest that feature analysis can be facilitated by decomposing a vector field into a divergence-free, a rotation-free and a harmonic part [PP00].
4.8 Tensor Icons

Tensor fields are a common quantity in engineering and physical sciences. Of particular interest are second-order tensors

\[ T = (T_{ij}) \]

which can be interpreted as linear transformations between vectors and are represented in 2D and 3D by \(2 \times 2\) and \(3 \times 3\) matrices, respectively. Examples are stress and strain tensors (see section 2.3), diffusion tensors (see subsection 3.3.2) and velocity gradients and viscous-stress tensors common in CFD.

This section concentrates on the visualization of symmetric second-order tensors which are characterized by \(T_{ij} = T_{ji}\) and are represented by symmetric matrices. Asymmetric second-order tensor fields can be visualized by decomposing the tensor into a symmetric and an antisymmetric part [DH93]

\[ T = \frac{T + T^T}{2} + \frac{T - T^T}{2} \] (4.3)

The first term is a symmetric tensor and the second term is an antisymmetric tensor which can be represented by an axial vector [DH93].

The meaning of these components can be explained by using an example from the field of computational fluid dynamics [Alo98]. Decomposing the velocity gradient tensor as shown above results in a symmetric and an antisymmetric component. The diagonal terms of the symmetric component describe the elongation of an infinitesimal fluid element in the coordinate directions while the off-diagonal components describe shear deformations. The off-diagonal components of the antisymmetric part describe the rotation of a fluid element (i.e. the vorticity of the fluid flow). A symmetric velocity gradient tensor represents therefore an irrotational vector field [Ken04].

For the remainder of this section the term “tensor” refers to a symmetric second-order tensor. Visualization techniques applicable to general second-order tensors are mentioned explicitly. We do not consider the visualization of higher-order tensors (see for example [KGM95]).

4.8.1 Tensor Glyphs

A popular way to visualize tensors is by depicting their eigenvalues and eigenvectors (see subsection 2.1.1). This can be achieved by drawing the eigenvectors as line segments whose length is proportional to the corresponding eigenvalues. In general the 3D perception of this representation is poor and using several of these glyphs simultaneously often leads to visual cluttering.

An improved representation is achieved by using tensor ellipsoids which encode the eigenvalues and the eigenvectors of a tensor by the directions and lengths, respectively, of the principal axes of an ellipsoid. Solid tensor ellipsoids can occlude large areas of a visualization. While this effect improves depth perception and decreases
visual cluttering it is often desirable to reveal a higher proportion of the field. An effective “see-through” ellipsoid is defined by using bands along the “equators” of the ellipsoids [PvPS95]. The “equators” of an ellipsoid are the intersections of the ellipsoid with the planes spanned by any pair of its principal axes. Consequently a 3D ellipsoid has three equators.

One disadvantage of tensor ellipsoids is the inherent perceptual ambiguity. For example, even if illumination is used it is difficult to distinguish between a sphere and a flat face-on ellipsoid [WMK+99]. Westin et al. propose an alternative tensor icon consisting of a sphere, a disk, and a rod with a common centre and where the diameters or length of these components are given by the minimum, medium, and maximum eigenvalues, respectively [WMK+99].

Haber represents a tensor by a cylindrical shaft and an elliptical disk [HM90, Hab90]. The colour and the length of the shaft indicate the sign and the magnitude of the maximum eigenvalue, respectively, whereas the radii of the disk represent the medium and minimum eigenvalue. The disk is coloured according to the magnitude of the maximum eigenvalue. Alternatively the disk’s colour can indicate the sign of the transverse eigenvalues as shown in figure 4.14.

Figure 4.14. Tensor Glyphs used in the stress analysis of crack propagation (©1993 IEEE Computer Society Press [KK93]).

If a visualization uses multiple tensor glyphs the icons must be scaled according to the biggest field value. Otherwise icons representing large eigenvalues might hide information and cause visual cluttering. For small tensors this leads to sparsely spaced icons and the visual connection between values is lost [LAKR98]. A solution is to normalise the icons with the maximum eigenvalue which creates a texture like
4.8 Tensor Icons

appearance for densely spaced icons [LAKR98].

4.8.2 Integral Representations

The previous subsection explained that using multiple tensor glyphs often causes visual cluttering. Perception is improved by replacing discretely spaced icons with a continuous representation such as hyperstreamlines [DH93, HD94]. The trajectory of a hyperstreamline is a streamline of an eigenvector field. The other two eigenvectors and corresponding eigenvalues define the directions and lengths, respectively, of the axes of the ellipsoidal cross section of the hyperstreamline. The hyperstreamline can be colour mapped with the eigenvalue which corresponds to the eigenvector defining its trajectory. Other scalar quantities can also be used.

An interesting alternative usage of colour has been suggested for the visualization of stress tensor fields. Delmarcelle and Hesselink discriminate between compressive and tensile stresses in the cross section of a hyperstreamline by associating colour with the angle between a radial vector of the cross section and the stress vector on the surface orthogonal to this vector. Angles of zero, 90, and 180 degrees correspond to pure tension, pure shear and pure compression, respectively [DH93].

An example of hyperstreamlines is given in figure 4.15. The image depicts four hyperstreamlines in the direction of the maximum principal stress in a solid under a point load. It can be seen that the maximum principal stress is roughly radially oriented around the point load with the transverse stresses increasing towards it.

![Figure 4.15. Hyperstreamlines and deformation surfaces visualizing the stress in a solid under a point load (indicated by the red arrow) (©1998 IEEE Computer Society Press [BP98b]).](image-url)
Non-symmetric tensors can be visualized by decomposing them into a symmetric and an antisymmetric component (equation 4.3). The symmetric component defines a hyperstreamline whereas the antisymmetric component defines a rotation vector depicted by a ribbon along the hyperstreamline.

Different approaches exist to compute the trajectory of a hyperstreamline. The straightforward solution is to use streamline tracking techniques for vector fields as introduced in subsection 4.7.3. In some instances this approach does not yield the correct structure of the underlying tensor field and better results are obtained by utilizing the full tensor information. Examples are found in the field of medical imaging where streamlines are propagated in diffusion tensor fields using Markov chains [PMF+98, PCF+99] and advection-diffusion mechanism [WKL99] (see subsection 3.3.4).

4.8.3 Surface and Volume Representations

The methods introduced so far represent tensor information at a point or along a streamline. Several algorithms exist to visualize tensor information over a surface or a volume.

Jeremić et al. connect (the trajectories) of hyperstreamlines originating from points on an open or closed curve with polygons and call the resulting glyph hyperstreamsurface [JSF+02].

Laidlaw et al. display a planar section of a 3D diffusion tensor field using techniques from oil painting [LAKR98]. The (planar) direction and the saturation of a brush stroke encode the 3D direction of the maximum eigenvector whereas stroke frequency, transparency and length-width ratio represent the maximum eigenvalue and various derived measures such as the diffusion anisotropy (see subsection 3.3.3). The authors claim that the image displays data at different levels of abstraction if viewed from different distances.

Boring and Pang visualize tensors by applying them to idealized objects, such as simple surfaces and volumes, which are subsequently displayed [BP98b, Bor98]. The tensor field manifests itself in the deformation of the object which includes twisting, bending, and elongation (see figure 4.15). In order to get meaningful results the authors decompose a tensor into a deviator and an isotropic tensor (see section 4.3).

Hagen et al. [HHW94] visualize deformation tensor fields by displaying their characteristics, such as points of minimum and maximum deformation, using generalized focal surfaces and characteristic curves.

Zhang et al. [ZCML00a, ZCML00b] visualize regions with large maximum and medium eigenvalues by surfaces spanned by the corresponding eigenvector directions. The resulting surface is at any point orthogonal to the direction of the minimum principal eigenvector.

Finally Kindlmann and Weinstein modify the opacity and colour transfer functions used in direct volume rendering in order to represent tensor quantities [KW99]. The technique is tailored for the visualization of diffusion tensor fields in the brain and was explained in more detail in subsection 3.3.4.
4.8 Tensor Icons

4.8.4 Tensor Field Topology

As with vector fields the complex structure of a symmetric second-order tensor field can be represented by its topology. In two dimensions the topological skeleton consists of degenerate points which are connected by separatrices. Degenerate points are points for which both eigenvalues of the tensor are equal and hence every vector is an eigenvector. Since the eigenvectors of a symmetric tensor are otherwise orthogonal degenerate points are the only points in the tensor field where the trajectories of an eigenvector field can cross. Separatrices are trajectories of the major eigenvector field connecting degenerate points and separating it into regions of similar behaviour.

The computation of degenerate points and the connecting separatrices is described by Delmarcelle and Hesselink [DH94, Del94] as follows: Let $T$ be a two-dimensional symmetric second-order tensor. A degenerate point $x_0$ is efficiently located by using the conditions

$$T_{11}(x_0) - T_{22}(x_0) = 0 \quad \text{and} \quad T_{12}(x_0) = 0$$

The partial derivatives

$$a = \frac{1}{2} \frac{\partial(T_{11} - T_{22})}{\partial x}, \quad b = \frac{1}{2} \frac{\partial(T_{11} - T_{22})}{\partial y},$$
$$c = \frac{1}{2} \frac{\partial T_{12}}{\partial x}, \quad d = \frac{1}{2} \frac{\partial T_{12}}{\partial y}$$

are then used to define the invariant quantity

$$\delta = ad - bc$$

which, if $\delta \neq 0$, defines the index of the degenerate point as

$$I = \frac{1}{2} \text{sign}(\delta) = \pm \frac{1}{2}$$

The index characterizes the streamline pattern of the major eigenvalue field in the vicinity of the degenerate point. It is possible to divide this pattern into sectors separated by separatrices. The separatrices are determined by the real roots $x_k$ of the cubic polynomial

$$dx^3 + (c + 2b)x^2 + (2a - d)x - c = 0$$

where $k = 1, \ldots, n$ and $n$ is the number of real roots of the above cubic polynomial.

The angle $\theta_k$ of the separatrices with the x-axis is given by $x_k = \tan \theta_k$ [Del94]. Only two types of elementary degenerate points exist as shown in figure 4.16: wedge points and trisectors. The wedge point in the figure has two different separatrices but it is also possible to have wedge points with only one separatrix and wedge points with three different separatrices in which case the middle one is ignored.
In time dependent flows pairs of trisectors and wedge points can merge to create patterns similar to vector field singularities (see figure 4.11). The type of the resulting pattern is dependent on the index of the degenerate points, e.g., two merged trisectors have a combined index of one and therefore look like a saddle point. Merged singularities are generally unstable and soon dissolve into wedges and trisectors [DH94].

In three dimensions a symmetric second-order tensor has three real eigenvalues. If only two of them are identical the same patterns as in 2D can be observed on a plane orthogonal to the third eigenvector. If all three eigenvalues are identical separatrices can be computed numerically from a non-linear system of equations derived from a spherical coordinate representation of the tensor [HLL97]. Each pair of separatrices forms a separating surface which is either a hyperboloid or a paraboloid.

In three dimensions degenerate points can appear along continuous lines and surfaces. Lavin et al. [LLH97] compute these points as the solution of an implicit function by using the singularities of the deviator of the tensor (see section 4.3). The authors further demonstrate that the singularities of the deviator can be linked to physical properties of the tensor field, e.g., for a stress field they represent the area with no stress, whereas for a deformation tensor field the singularities represent the area of the vortex core in the field.

4.9 The Classification of Visualization Icons

In this section we present a classification of visualization icons. Contextual elements and other tools increasing the effectiveness of a visualization are described separately in section 4.11.

The previous sections showed that today’s user can choose between a multitude of different visualization techniques. Classifying these techniques improves their understanding and comparison. The literature offers a wide variety of taxonomies for visualization algorithms and visualization results. Card and Mackinlay [CM97]
extend earlier work from Mackinlay [Mac86] and Bertin [Ber83] and classify the components of a visualization into marks (point, line, area, surface, volume), their graphical properties such as position (x, y, z-coordinate, time), retinal encoding (colour, shape, size, gray-level, texture and orientation), connections and enclosure, and textual elements.

Alternatively usability criteria can be used and a classification of 3D vector field visualization techniques according to these criteria is given in [MCG94].

In this thesis we classify visualization icons according to their type, spatial domain, and information scope. The classification extends a framework originally introduced by Delmarcelle and Hesselink [Del94, DH95] but is more comprehensive and extends to scalar icons as well. In addition several categories that were empty in the original work are now occupied indicating new classes of visualization algorithm introduced in recent years. The remaining empty categories show avenues for new research.

The type of an icon describes the type of a field represented by it. We consider scalar, vector and tensor icons. The spatial domain of an icon can be a point, line, surface, or volume and describes the dimension of the region for which field values are represented by the icon. For example, a vector arrow defines information for a single point whereas a streamline encodes vector information at each point of its trajectory. The embodied information scope is elementary if the icon represents the data only across the extent of its spatial domain; local if in addition derivative information (e.g., gradient, curl) is displayed; and global if the icon represents the structure of the whole field.

Furthermore we can define for each visualization icon the main visual attributes used for encoding the represented data. The following symbols are used:

- **P** Position
- **L** Length
- **D** Direction (angles)
- **V** Volume
- **C** Colour
- **S** Shape
- **T** Texture

The resulting classification is shown in the tables 4.2, 4.3 and 4.4. The main visual attributes for each icon are indicated in brackets. References to less well known icons are given in the footnotes of each table. The technique “Anisotropy Modulated LIC” refers to an extension of Line Integral Convolution for the visualization of diffusion tensor fields presented in chapter 7.

Some explanatory notes are necessary to understand the assignment of the principal visual attributes for each icon. For example, at first sight it might seem strange
that “position” is a visual attribute used by isocontours and ridge and valley lines, but not used by streamlines. However, the position information of a point on an isocontour or ridgeline gives information about the represented field: a point on an isocontour has a known isovalue and a point on a ridgeline indicates a field maximum in the transverse direction to the ridge line (i.e., both the visual attributes point and shape are utilized for perception). In contrast, the position of a single point on a streamline contains no information about the represented vector field: The vector direction is represented by the tangent at that point (i.e., the streamline’s shape) and the vector magnitude is usually represented by the point’s colour.

In order to demonstrate how the classification was achieved consider an “isosurface” which is a scalar icon. The spatial domain of an isosurface is a surface since each surface point represents a field value equal to the isovalue. In addition the isosurface represents local (neighbourhood) information in the sense that all points above and below the surface have field values above and below the isovalue, respectively. Analogously to isocontours the position of a point on the isosurface gives information about the values of the represented scalar field and the surface’s shape indicates the direction of the scalar field gradient. Hence an isosurface uses the visual attributes “position” and “shape”.

<table>
<thead>
<tr>
<th>Spatial Domain</th>
<th>Elementary</th>
<th>Information Scope</th>
<th>Global</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point</td>
<td>Discretized Height Fields (P or L) (e.g., Line and 2D Bar Charts) Color Mapped Particles (P,C)</td>
<td>Isocontours (P,S), Ridge and Valley Lines (P,S), Gradient Lines (S)</td>
<td>2D Scalar Field Topology (P,S)</td>
</tr>
<tr>
<td>Line</td>
<td>Colour Mapped Wireframe (C)&lt;sup&gt;a&lt;/sup&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Surface</td>
<td>Colour Mapped Surfaces (C), Height Fields (P,S)</td>
<td>Isosurfaces (P,S)</td>
<td></td>
</tr>
<tr>
<td>Volume</td>
<td>Direct Volume Rendering (C), Maximum Intensity Projection (C)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<sup>a</sup>see subsection 5.8.1

Table 4.2. Scalar icons.

It can be seen that almost all elementary higher dimensional icons employ visual attributes with a high perceptual dimension (shape or texture, often in conjunction with colour). In contrast elementary scalar data is usually represented by positional or colour information alone. Note that some icons, e.g., a height field over a planar surface, can be easily extended by a scale which creates a highly accurate representation of the data (since that way every surface point represents a position on a scale). Many icons for scalar data use colour which has a low representational accuracy. The reason for this is that colour also has a low spatial requirement which is ideal for representing continuous data over 2D and 3D domains. Continuous representations facilitate the detection of structure in the data and reduce visual cluttering
### 4.9 The Classification of Visualization Icons

#### Information Scope

<table>
<thead>
<tr>
<th>Spatial Domain</th>
<th>Elementary</th>
<th>Local</th>
<th>Global</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Point</strong></td>
<td>Arrow Plots (D), Particle Advection (S)</td>
<td>Critical Point Glyphs(^a) (S), Flow Field Probe(^b) (S)</td>
<td></td>
</tr>
<tr>
<td><strong>Line</strong></td>
<td>Streamlines (S,C), Illuminated Field Lines(^c) (S,C)</td>
<td>Stream ribbons (S), Streamtubes (S), Vortex Cores (S)</td>
<td></td>
</tr>
<tr>
<td><strong>Surface</strong></td>
<td>Stream surfaces (S,C), Bivariate Colour Maps (C), Line Integral Convolution (T,C)</td>
<td>Vector Field Texture(^d) (T,C), Global Line Type Features(^e) (S)</td>
<td>2D Vector Field Topology (P,S), Skin-friction Topology(^h) (P,S)</td>
</tr>
<tr>
<td><strong>Volume</strong></td>
<td>Direct Volume, Rendered LIC(^f) (T,C), Flow Volumes (S)</td>
<td>Vortex Tubes (S), Extremal Surfaces(^g) (P,S)</td>
<td>Skin-friction Topology(^h) (P,S), 3D Vector Field Topology (P,S)</td>
</tr>
</tbody>
</table>

\(^a\)[Löf98]  
\(^b\)[dv93]  
\(^c\)[SZH97]  
\(^d\)[KML99]  
\(^e\)[PR99a]  
\(^f\)[RSHTE99, IG98]  
\(^g\)[TM98]  
\(^h\)[HH91]

Table 4.3. Vector icons.

<table>
<thead>
<tr>
<th>Spatial Domain</th>
<th>Information Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Point</strong></td>
<td>Tensor Ellipsoids (S), Tensor Glyphs(^a) (S)</td>
</tr>
<tr>
<td><strong>Line</strong></td>
<td>Tensor Field Lines (S,C), Hyperstreamlines (S,C)</td>
</tr>
<tr>
<td><strong>Surface</strong></td>
<td>Anisotropy Modulated LIC(^b) (T,C), Laidlaw’s Paint Technique(^c) (T,C), Tensor Warping(^e) (S,C)</td>
</tr>
<tr>
<td><strong>Volume</strong></td>
<td>Diffusion Tensor, Volume Rendering(^f) (C)</td>
</tr>
</tbody>
</table>

\(^a\)[Hab90, WMK\(^+\)99]  
\(^b\)[WL01b]  
\(^c\)[LAKR98]  
\(^d\)[DH94]  
\(^e\)[BP98b]  
\(^f\)[KW99]  
\(^g\)[HLL97]

Table 4.4. Tensor icons.
and the oversight of interesting features.

The next section explains additional issues that are important when selecting visualization icons. It is followed by a discussion of how multiple icons can be assembled to form an effective visualization and how a visualization can be efficiently explored.

4.10 Using Visualization Icons

While the visualization task is made easier by using predefined visualization icons it is often not clear which subset of the given data should be mapped onto which visual attribute of an icon. An additional difficulty is that perceptual interferences between visualization icons can occur when displaying multiple fields simultaneously. This section gives a number of guidelines to assist with the visualization task.

The mapping between variables and attributes is usually determined by the intended purpose of the icon. The following purposes are common:

- Display quantitative information
- Draw attention
- Show correlation

As discussed in the previous section quantitative information is best displayed by length and position and is therefore reflected in the shape of an icon. Examples are vector arrows and height fields.

Figure 4.17 shows a circular vector field visualized with multiple visualization icons using our toolkit introduced in the next chapter. The velocity direction is indicated by a LIC texture and by the direction of the vector arrows. While vector arrows give more precise directional information they can easily be misleading since there is no indication of which data point they apply to (see subsection 4.7.1). The vector magnitude is represented by the colour of the LIC texture (a poor representation, but continuous), by the length of an arrow (precise, but only available for selected data points) and by a height field which offers the most accurate representation since the boundaries of the domain can be used as a scale. Note that only the height field shows clearly that the vector magnitude increases linearly from the centre of the data set. None of the visualization icons displayed illustrates the global location of the vortex core or the out-of-plane motion.

The example illustrates that different visualization tasks may require different visualization icons. The content of the visualization should be driven first and foremost by the information that the scientist needs to communicate to his or her audience.

Attention can be drawn to a target by using bright or highly saturated colours, movement or change, and sharp boundaries [RL95]. Target identification is also influenced by linear separation, colour category, and colour distance [Hea96]. If the complexity of the scene allows it, instant target identification can be achieved by using preattentive features.
4.10 Using Visualization Icons

Finally it has been suggested that correlation between related data sets is perceived most easily when similar visualization icons are used [KK93].

An additional way to classify visualization icons has been suggested by Bertin [Ber81] who distinguishes between two distinct uses of graphics:

- communicating some information which is understood.
- graphical processing in order to solve a problem.

The different results of selecting visualization icons according to usage are demonstrated by taking figure 4.17 as an example: If the user is interested in finding interesting flow features the LIC texture is suitable since it shows the entire vector field. In contrast, if the user wants to communicate selected results, say the direction and magnitude of the local flow maxima, then a more accurate and more easily understood icon such as a set of vector arrows might be more appropriate.

Several other issues have to be considered when employing visual attributes. Colour is a complex attribute due to the non-linearity of human colour perception and psychological influences such as colour metaphors. If colour is used in the segmentation of a scene no more than five colours should be used [Dav91]. Pastels should be used to show continuities (since they blend into each other) and clashing colours to discriminate areas [KK93]. Also colour should suggest meaning
Scientific Visualization

... metaphor) and related colours should be used for clusters of similar values or in a series of images [Cox88]. Colour discrimination just by hue is difficult [GL95] so if features are emphasized different saturations should be employed. However, when using illuminated coloured surfaces this would interfere with colour changes due to shading so that in this case colour spectra with constant saturation and brightness are preferred. Gray scales minimise complexities due to psychological influences and are especially popular in medical imaging due to their greater range of contrast compared with hue-only colour scales [KK93].

Textures are used to create visual richness without adding geometry. Van Wijk suggests that texture in data visualization is best used to symbolise global and quantitative information rather than local and qualitative information [van91]. Treinish is quoted in [KK93] as saying that the eye is more responsive to changes in texture than colour. Textures are therefore useful as a redundant cue for shape discrimination. Healey reports that texture and colour can only be combined if the texture has a strong textural salience [HIR99].

The shape of an icon can encode multiple dimensions. In the simplest case each dimension encodes one variable by elongation (e.g., tensor ellipsoids). Additional information can be represented using other shape related attributes such as curvature and “bumps”. The fact that rotated unfamiliar shapes are perceived as different indicates that icons encoding directional information should be simple and familiar to the audience. Shape perception can be improved using lighting, lightness and colour differences, texture, shadows, and contours (both explicit and subjective ones). The principles of Gestalt perception might be important in the design of visualization icons since it has been shown that well-organised good figures in the Gestalt sense are more easily remembered and make fewer demands on cognitive resources [Sch96].

Note that the visualization icons listed in tables 4.2-4.4 were restricted to scalar, vector, and tensor data, respectively. Some applications might require more specialised representations. Examples are uncertainty visualization and comparative visualization.

Uncertainty visualization aims to display uncertainty in scientific data due to errors introduced during measurement, simulation, transformation and interpolation. The goal is to avoid erroneous conclusions when interpreting the displayed data set. Uncertainty has been visualized by animation, sonification, psychovisual approaches, and by modifying geometry and attributes of existing visualization icons [PWL97, WPL96]. An interesting example has been suggested by Hin and Post [HP93] who add stochastic perturbations to particles paths in a fluid flow. Dense clusters of particles indicate high confidence in the simulated fluid flow in those regions.

Comparative visualization includes a range of techniques to visualize differences between two data sets. Examples are the comparison of data sets of healthy and diseased organs in medical imaging and the comparison of simulated data with experimental data. An example for this application is the validation of models in engineering [PP95, SPU98]. Comparison can be achieved on three different levels. Data level comparison techniques compare raw data directly using various metrics...
and derived data sets. Differences can then be visualized. \textit{Image level comparison} is achieved by taking images, usually obtained as the result of a visualization, as input and by comparing them using difference images, Fourier analysis or overlay and layout techniques [PP95]. Finally \textit{feature level comparison} uses extracted features as input. An example is the comparison of vector fields by means of their singularities [LBH98, BH99].

\subsection{Combining Visualization Icons}

Visualization icons can be combined so as to yield more information than the sum of the individual icons [KK93]. Additional information may exist in the form of correlation between multiple variables or as higher-order visual information (Gestalt). Correlation between related data sets is perceived most easily when similar visualization icons are used [KK93]. Perception can be further improved by using multiple visualization techniques simultaneously for the same data [RL95].

Gestalt concepts in visualization are demonstrated by Laidlaw et al. who use densely-arranged normalised tensor ellipsoids in order to obtain a texture-like representation of a diffusion tensor field which improves the perception of features and field properties [LAKR98].

In contrast to correlated variables unrelated variables are best displayed using orthogonal (independent) visual attributes such as shape, colour, movement, and texture. Many visualization icons utilize multiple visual attributes so extra care has to be taken when combining such icons. In general it has been shown that the brain can handle a maximum of about seven unrelated elements [KK93]. Note that different visualization icons can also be used to display the same data in order to reinforce information or to highlight different aspects of the data (explicit redundancy). An example of using orthogonal visual attributes is shown in figure 4.18.

Further guidelines for combining visualization icons are obtained from research on graphing data. For example, visualization icons which overlap should be visually distinguishable [Cle85]. When visualization icons with similar shape are used colour can be used to discriminate between them [Cle85]. Mackinlay [Mac86] extends work from Bertin [Ber83] and classifies graphical encoding techniques into marks (points, lines, areas), positional (1D, 2D, 3D), temporal (animation), retinal (colour, shape, size, saturation, texture and orientation), maps, connections (tree, network) and others. The author uses this classification to create a composition algebra which specifies whether two encoding techniques can be used for the same task. While the work was developed for graph design many results apply to the field of scientific visualization. For example, when size and shape are composed together small sizes must be avoided since the shapes of the small objects may be hard to distinguish.

This observation is reflected in our classification of visual attributes given in table 4.1. For example, length has a medium spatial requirement but shape has a high spatial requirement. Consequently an icon using both length and shape for encoding data is restricted in its usage by the spatial requirements for shape encoding. In general the spatial requirement of an icon is given by the largest
spatial requirement of any of its visual attributes.

The effectiveness of visualization icons is also influenced by the chosen background. The background can be used to highlight and support features in the image and can be used to provide supplementary information and 3D perspective [KK93]. Keller and Keller recommend that the background of a visualization should have a neutral (unsaturated) colour with a good contrast to the foreground [KK93]. The authors further recommend the use of a horizontal (landscape) view since it corresponds to the normal field of vision. 3D scenes should be oriented in such a way that important features are in the foreground and not covered by other scene components [KK93].
4.11 Increasing the Effectiveness of a Visualization

A general approach for the creation of effective visualizations is given by the “Natural Scene Paradigm” which is based on our ability to immediately perceive complex information in natural scenes [Rob91]. Implementing this paradigm involves clear 3D structures and the association of data with recognizable properties of objects.

In many cases no natural association between data and icons exist and the understanding of a visualization is dependent on the target audience having a priori knowledge about it. In particular familiarity with the data set and the particular visualization techniques is often required.

4.11.1 Lighting

Since our environment is illuminated by a wide variety of natural and artificial light sources, the human brain is well adapted to perceive geometric information from shading. As mentioned in subsection 4.4.3 the single most important clue in shape recognition is diffuse illumination. Hence lighting is essential if using icons which encode information by shape (such as height fields, isosurfaces, and tensor ellipsoids). On the other hand, illuminating an object changes its perceived colour so that lighting should be disabled if colour is the primary visual attribute. For example, in this work lighting is disabled for the rendering of flat or nearly flat (“shape-less”) colour mapped surfaces.

Some authors suggest that diffuse shading is the most important shape cue and that adding specularity does not significantly improve perception of shape differences [RB00]. Therefore specular material properties should be avoided for icons which use colour as a secondary visual attribute. On the other hand specular highlights can help to distinguish object details, such as the radius of a rounded edge, so that adding specularity to objects with low colour variations (e.g., isosurfaces) improves the amount of perceived information.

The use of shadows can further improve the perception of the 3D geometry of an object. For example, shadows have been successfully employed for visualizing 3D vectors over 2D slices [KH91]. Shadows can also be used to indicate the distance of an object from a background plane and help to indicate the spatial order of objects. It has also been shown that shadowing increases the accuracy (but not speed) of object positioning [HWSB99]. However, Hubona et al. [HWSB99] show that using multiple shadowing light sources decreases user performance for positioning and resizing tasks, which indicates that the perception and interpretation of scientific visualizations might also suffer if more than one light source is used for shadow creation.
4.11.2 Perceptual Clues

In general it is difficult to design a scientific visualization using the natural scene paradigm. More concrete techniques for improving perception and understanding are

- Shape clues
- Contextual clues
- Annotations

Shape clues are used to improve the perception of the 3D geometry of a scene. Two major classes of shape clues exist: illumination (explained in the previous subsection) and explicit redundancies. Techniques based on explicit redundancies include emphasizing of silhouette curves (figure-ground boundary) and contour curves (depth discontinuities) [ST90] and the use of mirrors. Projections of coloured shadows on the 3 coordinate planes have also been used [Tuc97].

Contextual clues improve perception by enabling the brain to relate abstract visualization icons to familiar objects or properties. Examples of contextual clues inherent in a data set are coastlines, bounding boxes, and model outlines which improve the perception of positional information. Motion blur can be used to indicate velocities. Additional contextual clues to make data more readable include numbered scales, grid lines, and abstract objects to suggest value and relationships (see [Tuf83]). Object recognition can be increased by comparing an object with similar ones familiar to the user. This can be achieved by using multiple windows with the same view, by using split-screen techniques or by using overlay techniques.

Zhang et al. [ZCML00b] apply contextual clues to the field of medical imaging and use easily identified anatomical features to improve the understanding of the 3D geometry of visualized nerve fiber structures. An example from our own work [WL01b, Wün03b] is shown in figure 4.19. The tube-like structures indicate nerve fiber tracts, whereas the green and red isosurfaces represent the eyes and the ventricles, respectively. The latter two objects are anatomical landmarks which indicate the orientation of the data set (the eyes are in the front of the head), improve the perception of the position of the nerve fibers inside the head, and clarify the perception of the 3D geometry since the fibers tracts furthest away from the view point are occluded by the ventricles.

Finally annotations can be used to identify features and to explain relationships. Examples are legends, labels, and markers. Legends should be comprehensive, informative and draw attention to important features in the data set [Cle85]. Care has to be taken that the annotations do not distract from the actual goal of the visualization [Tuf83].

4.11.3 Exploration Techniques

The perception of a visualized data set is further improved by enabling the user to interact with the data. Common types of interaction are rotation, translation (pan)
and zoom. Walk through and fly through features are also popular. An example of
the resulting improvement in perception is explained in [Wüm03a]: If large numbers
of icons are distributed over a 3D domain, rotating the model around its axis enables
the brain to differentiate icons in the foreground and the background. Animating
the interaction, e.g., using continuous rotations or automatic fly-throughs, can help
the user to concentrate on the data.

Other common interaction techniques are fish-eye views and cut-away (clipping)
techniques. A generalisation of clipping is our 
sectioning tool
 introduced in the
following chapter. The tool slices a data set into sections and arranges them regularly
in 3D. Inner structure is revealed but the global structure can still be perceived due
to the brain’s ability to visually interpolate slice data.

Region-of-interest techniques allow the user to extract a region of interest from
the data volume. In most cases regions of interest are interactively defined by placing
a box or a sphere into the volume. In some cases simple shapes are not sufficient
to extract an interesting region such as an anatomical abnormality in a medical
data set. Ney and Fishman [NF91] present a tool for interactively creating shapes
suitable to define arbitrarily shaped regions of interest in a data set.
Fuhrmann and Gröller [FG98] introduce magic lenses and magic boxes as a tool to improve 3D interaction. Magic lenses are planar usually circular objects which magnify the scene behind the lens and remove the volume in front of it. Magic volumes are explicit focus volumes in which a detailed representation of the visualization is displayed. The back faces of the box are opaque in order to reduce distraction.

Large data sets can be efficiently explored by using multiscale visualizations which use different levels of abstraction for detail views and overviews of the data. An example is given by Stolte et al. [STH03].

Data brushing, originally developed for multivariate data, can also be utilized in scientific visualization by selecting or highlighting icons shown in one view in all other views of the data. The technique can be used to increase rendering speed, to highlight or to extract features and to facilitate the discovery of relationships between subsets of the data [WB96, WB97].

Finally image graphs [Ma99, Ma00] and visualization spreadsheets [CRBK98, JKM01] allow the user to interactively change visualization parameters while seeing their effects on the final image.

An increasingly popular approach to dealing with extremely large data sets is the use of immersive environments such as virtual reality (VR) workbenches [BL91, Bry96] and CAVE theatres [Jas97]. Interaction with data is achieved using data gloves [BL91, Bry96, FBZ+99] or natural interaction techniques such as speech and hand gestures [SZP+00].

For large data sets exploration results might be improved by employing a collaborative visualization in which research teams collectively analyze data [WWB97]. Fuhrmann et al. suggest that collaborative exploration is facilitated by using augmented reality which combines familiar physical surroundings with synthetic data [FLSG98]. Issues relating to collaborative control are discussed in [BIP00]. Recently collaborative visualization over the Internet has been proposed as an effective learning tool [Pea02].

Direct interaction with the data can be replaced or supplemented by a presentation simulating an interaction. For example, a sequence of successively magnified images reveals structure whereas a simultaneous display of images using different techniques shows multiple aspects of a data set [KK93].
The previous chapters illustrated that biomedical data sets can comprise a diverse range of measurements such as tissue densities, sensitivity to magnetization, blood flow velocity, and material strain. The size and complexity of these data sets makes it increasingly difficult to understand, compare, analyze and communicate the data. Visualization is an attempt to simplify these tasks according to the motto "An image says more than a thousand words".

This chapter introduces a toolkit developed for exploring and visualizing complex biomedical data sets with a particular emphasis on tensor fields. The program was written in C++ using OpenGL and FLTK, a LGPL’d C++ graphical user interface toolkit for X (UNIX), OpenGL, and WIN32 (Microsoft Windows NT 4.0, 95, or 98) [Spi].

The main contributions of this chapter are as follows: We suggest a modular toolkit design which facilitates the comparison and exploration of multiple data sets and visualizations. We introduce a novel field data structure which allows interactive creation of new fields and we present boolean filters as a universal visualization tool. Finally we explain how model properties can be computed from a finite element model and we suggest some new visualization techniques and several improvements to existing ones.

The chapter starts with a survey of popular visualization environments and motivates the development of our toolkit. An overview of the overall design of the toolkit is followed by an explanation of our novel field data structure. We then introduce tools for 3D interaction, tools for selecting and creating colour maps, and tools for selecting volumes, surfaces and points for visualization purposes. The following sections present the various visualization icons implemented in our toolkit and explain the computation of model properties from FE models. We conclude with additional remarks about the rendering control, input and output techniques, and advanced features for exploring and annotating data sets.
5.1 Visualization Environments

The main objective of this thesis is the visualization of tensor fields in biomedicine. As previously illustrated biomedical structures are often modeled using finite elements and a visualization tool for such structures should therefore be able to handle curvilinear grid data and data representations in world and material coordinates.

All popular visualization environments and programming frameworks known to us are general enough to cover a wide range of visualization tasks but can also be customized for specialized tasks. Visualization environments, such as AVS, IRIS Explorer, OpenDX (formerly IBM Data Explorer) and VTK contain modular and extensible components covering the entire visualization process from data input and transformation to rendering. The underlying visualization model (data flow paradigm) consists of three processes: Filtering maps data into data, mapping maps the resulting data into geometric primitives and rendering renders these primitives into images.

AVS/Express, IRIS Explorer, and OpenDX improve usability by providing a visual programming environment in which suitable processes are connected to form a visualization pipeline [AVS, NAG98, IBM97]. The applications incorporate most of the common visualization techniques and allow curvilinear grids. AVS also permits a variety of different coordinate systems [AVS] and allows incremental rendering and data references for efficiency improvement [Lor95]. Iris Explorer claims to be the first available commercial visualization software to support collaborative visualization [NAG, NAG98] and also incorporates image processing modules.

OpenDX [IBM] is an open source version of the IBM Data Explorer and differentiates itself by incorporating “smart interactors” which examine the data and create meta data and suitable option menus. An example is the vector interactor which reconfigures itself according to the dimensionality of the data and the maximum and minimum of each vector component. Other data-driven tools are the Color Map Editor and the Sequencer used for animations. OpenDX extends the data flow mechanism by introducing commands to store the state of a visualization which can be used to combine icons from different frames of a time varying data set. Efficiency is improved by computing only required results in the data flow graph [AT95a, AT95b].

One disadvantage of most commercial data flow based visualization environments is that for efficiency reasons graphic objects are “final” and can not be reused as input to other modules [FH94]. For example the isosurfaces produced by a polygonization module cannot be used as input data set for a 2D streamline algorithm. Note, however, that it is usually possible to write specialised modules which generate an isosurface of an appropriate input type format for further processing. A survey of various general and specialized packages is found in [Bra95] while Globus and Uselton suggests a standardised test suite for the evaluation of visualization software [GU95].

A couple of additional visualization environments deserve explicit mention: VTK is a public domain programming framework implementing the data flow paradigm [SML96a, SML96b, SAH00, Kit]. Applications can be built in Java, C++ or us-
5.1 Visualization Environments

ing the interpreted command languages Tcl/Tk or Python/Tk. Chen reports that
the latest version of VTK offers execution efficiency and visualization capabilities
comparable with commercial systems [Che99]. Current versions incorporate most
of the common visualization techniques including some tensor icons, such as tensor
ellipsoids and hyperstreamlines. A visual programming environment for VTK with
an integrated self-learning help capability has been proposed [Tv00].

Interactive Data Language (IDL) is a data manipulation language for data anal-
ysis and visualization [Res]. Its key features are image processing analysis routines,
integrated mathematical and statistical functions, GUI tools, and data mining tools
[Res]. IDL contains a library of colour map look-up tables and most of the basic
visualization icons, such as isosurfaces, isocontours, particle traces, streamlines and
colour mapped surfaces. Visualizations are created using a scripting language. Data
can not be defined over curvilinear grids [Res99]. An NCSA Training web page
recommends to avoid IDL when creating high quality visualizations and animations
[NCS].

Amira, originating from the Department for Scientific Visualization of the Konrad-
Zuse-Zentrum für Informationstechnik Berlin (ZIB), Germany, is an integrated 3D
visualization and volume modelling program for medicine, biology, and engineering
[ZIBa]. The program supports multiple coordinate systems, curvilinear grids, and
the creation of grids for FE simulations. Processing of 3D image data is supported
by automatic and interactive segmentation tools. Advanced vector field and volume
visualization tools are available and an extensible development version has been
released.

All of the above introduced visualization environments form an acceptable foun-
dation for a tensor field visualization workbench. Unfortunately, AVS, IRIS Ex-
plorer, Amira, and IDL are commercial software packages and OpenDX went open
source only recently. We considered using VTK as a framework for our thesis, how-
ever, when we commenced this work VTK had considerable less features than now.
Also VTK does not accept curvilinear grid data as input. We therefore decided
to develop a visualization toolkit from scratch using OpenGL [WND97] and FLTK
[ Spi], a public domain toolkit for GUI development. The results of this research are
presented in this chapter. Novel features which according to our knowledge are not
or only partially supported in the previously mentioned tools are explained in some
more detail.

While we refer to our application as a visualization toolkit it really represents
the prototype of an integrated visualization and modelling environment. The term
“toolkit” has its origin in our original idea to create a workbench for testing and
experimenting with tensor field visualization techniques.

It is interesting to note that recently an application has been published specifi-
cally for visualization diffusion tensor imaging data [Sci].
5.2 Top Level Design of the Visualization Toolkit

The top-level control of the visualization toolkit manages lists of models, visualizations, visualization windows, rendering controls, and colour maps which are used for the creation of visualization icons. Figure 5.1 shows a screen shot of the toolkit at work.

All models used in our toolkit are finite element models. If a data set is not associated with a FE model (e.g., MRI raw data) a single trilinear element representing the bounding box of the data volume is used as a default model.

![Figure 5.1. An example of the visualization toolkit at work. The image shows the toolkit control at the top-left, two visualization windows partially covered in the middle, two visualization controls at the left, and one rendering control in the top-right and the colour map control partially covered in the bottom-right.](image)

A visualization object contains a display list of visualization icons (high-level graphic primitives) and a set of transformation parameters. Most common visualization icons such as particle systems, vector glyphs, streamlines and streamtubes, hyperstreamlines, colour mapped surfaces, height fields, isosurfaces, tensor glyphs
and various types of line integral convolution textures are implemented. Annotations used to identify features and to explain relationships can also be created. Examples are legends, labels, and markers.

Transformations are necessary to align two data sets, e.g., MRI and PET data, or to represent two models, such as a healthy and a sick heart, at the same scale. A visualization object also contains data fields associated with the model (including any interactively defined new measures), field macros, and a list of boolean filters defined over the data fields. Additionally the visualization contains a list of volumes, surfaces and point sets of interest which among others are used to define the location of visualization icons.

A visualization window displays a visualization object with rendering parameters provided by a rendering control object. A rendering control contains a view, a trackball, lighting information, mirrors, and global clipping planes. The same rendering control can be used for different windows which is useful, for example, when comparing two models. Vice versa the same visualization can be displayed in different windows with different rendering parameters, for example, in order to display two different sides of the model simultaneously or in order to give a global and a detail view.

Each visualization is associated with exactly one model while a model can have several visualizations. The advantage of this design is that the user can simultaneously run two visualizations of the same model (e.g., from different research groups who examine the same data set) with different visualization icons and derived fields.

The colour map control manages a list of colour maps used by the visualization icons. The decision to provide “global” colour maps was motivated by users who found it easier to interpret visualizations when identical colour maps were used for the visualization of related data fields. A typical example is the comparison of the myocardial principal strain field in a sick and a healthy left ventricle.

The top-level control of the toolkit, shown in the top-left of figure 5.1, displays the relationship between its components graphically and allows the user to hide, show, add and delete components. The top-level class structure of the toolkit is displayed in figure 5.2 using a class diagram. All class diagrams used in this chapter were constructed using UML (unified modeling language) and the software package Rational Rose [Rat].

5.3 A Field Data Structure for Interactively Exploring Biomedical Data Sets

In order to explore a data set efficiently and effectively it is desirable to interactively choose and create new measures and different data representations. For example, the user might want to compare the difference in velocity magnitude or flow direction of two velocity fields. In this section we present a data structure and a user interface designed for that purpose.

Previous work related to this area seems to be extremely limited. Bryson et al.
Figure 5.2. Top-level class diagram of the visualization toolkit.
[BKGY96] describe a Field Encapsulation Library which provides a grid independent interface to gridded three dimensional field data. Moran and Henze [MH99] and a previous paper by us [WL99] suggest a tool where new fields can be created from existing fields using a list of predefined operators. The computation is efficient in the sense that the new field is computed only at data points where it is needed. However, only fields defined over the same grid can be combined. Moran and Henze prevent recomputation of the same value by caching vertex values once they have been created. Our work allows the combination of fields from different domains and permits a more powerful variety of derived fields. Furthermore, since we don’t interpolate sample values but use the representation of the underlying source field, the resulting field values are more accurate.

The following two subsections introduce first a general field data structure and then the graphical user interface used to create new fields and field macros. The capabilities of the data structure are demonstrated in the case studies in chapter 6 and 7.

5.3.1 Field Data Structure

In order to unify different field representations our toolkit represents the domain of the entire data set (the model) by a finite element mesh. Note that this prerequisite does not limit the range of input data sets since, for example, the domain of a mathematical function or of an MRI data set can be represented by a single trilinear interpolated cuboidal element enclosing the region of interest.

When combining fields specified in different coordinate systems (i.e., world coordinates \((x, y, z)\) or different material coordinates) a mapping between coordinate systems must be accomplished. The mapping from material to world coordinates is achieved by using the finite element interpolation. The reverse mapping requires a multi-dimensional Newton method [PVTF92]. We found that 3 iterations are usually enough to find a material point inside an element for a given point in world coordinates. The coordinate transformation of vector and tensor quantities has been described in subsection 2.4.3 on page 26.

Using these techniques it is possible to define fields with different domains and to combine them using operators. Our data structure consists of an abstract field class that is first subclassed into a (symmetric) tensor field, a vector field, a scalar field, and a general \(n\)-d field class as shown in figure 5.3. These classes contain attributes and methods common to their subclasses. For example, for every symmetric tensor field the algorithm to compute eigenvectors and eigenvalues is identical, and for every vector field it must be distinguished whether it is signed or unsigned. All of these classes are then subclassed into FE mesh fields, regular grid fields, derived fields, analytic fields, and expression fields.

Note that while multiple inheritance might sound like a more appropriate design technique it has several drawbacks [Eck02] and in our case does not significantly improve either code reuse or readability.

A FE mesh field is associated with a finite element mesh and a set of element
interpolation functions. The type of interpolation functions depends on the spatial variation and the continuity requirements of the field. In particular the interpolation functions for FE mesh fields are not necessarily the same as the ones used to interpolate the geometry of the underlying model. An example of a FE model which uses different interpolation functions for different fields is the numerical model of a canine heart introduced in subsection 3.2.8.

A regular grid field contains a regular 3D grid of sample values, a single trilinearly interpolated finite element which specifies the domain (coordinates) of the sample values, and a reconstruction filter used to interpolate the sample values.

A derived field is associated with a parent field and contains a function specifying how a field value is derived from the corresponding parent field value. As an example consider an eigenvalue field which has a tensor field as a parent. The eigenvalue field contains a link to the associated tensor field, a variable specifying whether the major, medium, or minor eigenvalue is selected and a method to compute the eigenvalue at a point. Other examples of derived fields are eigenvector fields (major, medium, or minor), vector length fields, vector angle fields (specifying the angle with any of the world or material axes), gradient fields, and vector and tensor component fields. For most FE models the user is interested in the components of a tensor with respect to the material coordinate system of the model so that a basis transformation is performed if the tensor is defined with respect to a different coordinate system (see subsection 2.4.3).

![Figure 5.3. Top-level class diagram of the field data structure.](image)

An analytic field is specified by an algebraic function defined over a domain in world coordinates or element coordinates. This type of field proves useful when creating test cases for visualization algorithms and can be used in applications where the analytic solution to a problem is known.

Finally an expression field contains an arithmetic expression tree where the leaves are numeric constants or are fields themselves.

Figure 5.4 demonstrates the subclassing of the field data structure in figure 5.3.
5.3 A Field Data Structure for Interactively Exploring Biomedical Data Sets

Figure 5.4. Class diagram of (a subset of) the scalar field data structure.

by showing a subset of the scalar field class hierarchy. Note that the computation of the gradient function is implemented in subclasses since the most suitable computation method depends on the type of a field. For a regular trilinearly interpolated sample grid finite differences can be employed, for analytic functions a numerical approximation can be used and for higher-order finite element meshes the derivatives of the interpolation functions can be used to get the coordinate derivatives of the field.

The advantages of our field data structure compared to precomputing field values at common sampling points are threefold:

- we eliminate problems with the interpolation of derived values. For example, directly interpolating the eigenvalues of a tensor over a finite element gives usually the wrong results. Instead we interpolate the tensor and compute the eigenvalues from the resulting tensor.

- we can combine arbitrary fields through arithmetic functions (e.g., the difference between two scalar fields) even if they are defined over different grids. Similarly, we can interactively derive new fields by choosing a parent field for a derived fields.

- No additional sample errors are introduced as would happen, for example, if
The Visualization Toolkit

sampling an analytic field over a fixed grid structure and then interpolating those values.

- Entities defined over a finite element grid can be represented with respect to either the world coordinates or the material coordinates. This choice of representation increases the power of the visualization. For example, chapter 3 demonstrated that some entities defined with respect to material coordinates, such as the circumferential strain, are more meaningful for diagnosis.

The disadvantage of the described field structure is that

- the computation of a derived field value is slower than if the field values were precomputed at sample points.

5.3.2 Graphical User Interface

We have implemented a graphical user interface which allows the interactive derivation of new (expression) fields from existing fields. Figure 5.5 shows on the left the user interface for managing fields and the visualization of a model. The right hand side of the figure shows the user interface for creating a new field. The three output text components on the top contain the currently defined scalar, vector and tensor fields. The user can create a new field by inputting a simple mathematical expression which currently can contain the following elements:

Scalar field: Expressions for selecting eigenvalues and components of tensors, components of vectors, numerical constants, binary operators (+, -, *, /, ^), unary operators (sin, cos,...), vector length, trace, angle of a vector with the $x$, $y$, $z$, $\xi_1$, $\xi_2$, or $\xi_3$-coordinate axis.

Vector field: Expressions for selecting eigenvectors of a tensor, gradient of a scalar field, binary operations (+, -, cross product), vector constants.

Tensor field: Binary operations (+, -, *), tensor constants.

We have also implemented a conditional expression

\[
\text{switch}(<\text{cond1}>:\text{field1};\ldots;\text{default}:\text{fieldN})
\]

Currently the conditions are restricted to boolean expressions containing scalar fields and comparison operators only. Chapter 7 demonstrates how such an expression can be used for the visual segmentation of an image.

The left part of figure 5.6 shows a visualization obtained by using the field defined in figure 5.5. If the user is not satisfied with the result the expression field can be edited in the modification window shown on the right of figure 5.6. Using the update button of the visualization control (figure 5.5 left) the user can recompute any visualization icons dependent on that field.

Expression fields also offer a convenient way to create visualizations for multiple versions of a field $F$. In order to do this define a new field $E$ equal to one version
5.4 User Interaction in 3D

Figure 5.5. A visualization control (left) and the graphical user interface for creating a new field (right).

of the field $F$ and use $E$ to derive other fields which are then visualized. If we want to visualize a different version of $F$, say $F'$, it’s sufficient to set $E$ equal to $F'$ and to update all visualization icons. This property is useful, e.g., when comparing visualizations for raw and smoothed versions of the same data set.

Frequently used expressions can also be saved as a macro and the macro name can then be used during field creation. An example is given in figure 5.7.

5.4 User Interaction in 3D

Two types of user interaction are essential in a visualization toolkit: visualization of the model from different view points and selection of and interaction with objects in the visualization domain.

5.4.1 View & Model Transformations

The visualization of a scene from different view points is achieved by using the standard OpenGL camera model and by implementing rotation, translation, and
Figure 5.6. The visualization of the field defined in figure 5.5 (left) and the user interface used to edit an expression field (right).

Figure 5.7. Defining a macro and using it to create a new field.
5.4 User Interaction in 3D

Zooming. The camera is specified by an eye point, a view direction and an up-vector [WND97]. The camera is translated parallel to the view plane by moving the mouse and pressing the <alt>-key and the left mouse button. The perceived movement is that of the scene being dragged with the mouse.

Rotation is performed using a virtual trackball [BY96]. Moving the mouse cursor in the visualization window while pressing the left mouse button is interpreted as movement on the surface of a virtual sphere enclosing the window. The model is rotated according to the mouse position on the virtual sphere. The trackball can be animated (i.e., it keeps spinning after releasing the mouse button) by enabling the corresponding button in the rendering control associated with the window. The speed of the rotation depends on the speed of the mouse movement. An alternative to the trackball is the arcball [Sho94] which has the advantage that the defined rotation matrix is not path dependent and more versatile (i.e., constraint rotations about an arbitrary axis can be implemented). However, we found that trackball rotations are more intuitive and therefore use them for our toolkit.

Two different functions for zooming are provided. By pressing the <ctrl>-key and the left mouse button the user can select a rectangular area of the current window which becomes the new view of the scene. Using a selection rectangle with a shape different from the display window leads to a distorted display. This can be avoided by enabling a fixed viewport ratio in the rendering control.

Zooming can also be achieved by moving the mouse while pressing the <shift>-key and the left mouse button. The zoom factor is determined by the ratio of the distances of the start point and current point of the mouse movement to the centre of the visualization window. A mouse movement from the window’s centre to its corner increases the model size by a factor of three, whereas the opposite movement decreases it to a third of its original size. Smaller movements correspond to smaller zoom factors.

Note that rotation, translation and zooming of the model are implemented by the rendering control object (see section 5.11) and are therefore identically applied to all windows using the same rendering control.

5.4.2 3D Object Selection and Interaction

In order to interactively place, transform and modify objects, such as regions of interest or annotations, it is necessary to implement an object selection mechanism. The toolkit uses the standard OpenGL selection mechanism [WND97]. For example, in order to select a point with the mouse all selectable points are drawn in the GL_SELECT mode while loading a unique name for each point. OpenGL returns a list of primitives that intersect the viewing volume specified by the current model-view and projection matrix. In order to pick a point with the mouse it is necessary to define a pick matrix using the command gluPickMatrix with the current mouse position as an argument. All objects rendered onto the viewport region specified by the pick matrix are then stored in a selection buffer which can be examined to find the front most point hit.
In many cases, such as the specification of a region of interest, we want to translate or scale objects in 3D. This is achieved by specifying for an object translation handles and modification handles which are rendered as blue and red points, respectively. Translating a translation handle translates the whole object, whereas translating a modification handle only changes that point and as such deforms the object. Figure 5.8 illustrates these concepts.

![Figure 5.8](image)

**Figure 5.8.** A cube object with a translation handle (blue) and a modification handle (red). Part (b) of the figure shows the scene from part (a) after moving the blue handle which translates the cube object. Part (c) of the figure shows the deformed object obtained by moving the red handle.

Handles can be selected and translated in 3D by interpreting the mouse movements on the screen as an object movement parallel to the view plane. This is achieved by determining the z-buffer value of the selected object. The new mouse position is then projected back into the 3D domain so that the z-buffer value stays constant. An example is given in figure 5.9. Note that the object is moved in world coordinates \((x, y, z)\) which in general are not axis-aligned with the view coordinates \((u, v, n)\).

In many engineering applications objects are moved in 3D by moving them parallel to the coordinate planes which is achieved by using orthographic projections in the direction of the three world coordinate axis. Our method is a generalization of this, but if required an axis aligned view is obtained by pressing the ‘x’, ‘y’, or ‘z’-key, which aligns the \(yz\), \(xz\), or \(xy\)-plane of the world coordinate system, respectively, with the \(uv\)-plane of the view coordinate system. Perspective and orthographic projections can be selected from the rendering control.

In order to improve 3D understanding of the scene and to make positioning of objects easier the user can split a window into four view ports by pressing the ‘s’-key. The bottom-right view port shows the scene with the current view whereas the top-left, top-right and bottom-left view port use a view along the \(z\), \(y\), and \(x\)-axis of the coordinate system, respectively\(^1\). An example is given in figure 5.10.

\(^1\)Actually rather than changing the camera we rotate the model such that the \(x\), \(y\) and \(z\)-axis are aligned with the current view direction. The two operations are equivalent.
5.4 User Interaction in 3D

Figure 5.9. An object is moved in 3D by converting a mouse movement on the screen to a movement in world coordinates \((x, y, z)\) parallel to the screen.

Figure 5.10. Manipulating an object in 3D is facilitated by splitting a window into four parts and by displaying simultaneously the current view (bottom-right) with projective views aligned with the three coordinate axes.
Note that the trackball operation is only applied to the bottom-right window, whereas zooming and translation are applied to all view ports. Return to the full screen view is achieved by pressing the ‘u’-key.

Alternative methods for 3D point manipulation include arcballs and two pointer input [PGV99, ZFS97] but are not (yet) implemented. A survey of 3D interaction techniques is given in [Han97].

5.4.3 The Field Probe

Using the previously described principles we have designed a field probe which the user can move through the 3D domain in order to explore the data set. At any point the probe shows the current world coordinates. If the probe is inside the current model the element ID, material coordinates and the material coordinate directions at the probe’s position are shown. Additionally, the values of all enabled fields at that point are output (see figure 5.11).

Note, that since the probe is moved in world coordinate a multi-dimensional Newton method [PVT92] must be employed to find the corresponding material coordinates. When computing the material coordinates for the start point of the probe and during sudden large fast probe movements a brute-force method is used which tests all finite elements for the current world coordinate point. In all other cases the start point of the search for the material coordinate of the probe is given by the element number and the \( \xi \)-coordinate of the last position of the probe. If the returned \( \xi \)-coordinate lies outside the element the algorithm determines the face intersected by the line connecting the old point and new point in parameter space. By using a precomputed adjacency matrix the neighbouring element is obtained and the search continues using the intersection point with the face as the start point of a new Newton search. We found that on average three iterations are sufficient to determine the material coordinates of the probe when moving it through the model.

The Newton method fails in the vicinity of degenerate points in the underlying FE model (i.e., regions where finite elements have two or more duplicate vertices). The reason for this is that the Jacobian of the element interpolation functions (equation 2.25) is not invertible at such points. An example of a degenerate point is the apex of the heart models (see subsection 3.2.8).

One possible solution is to use a singular value decomposition (SVD) to isolate the singular terms in the Jacobian matrix and to use a lower dimensional Newton search (this technique is used in NASA’s Plot3D code). A cruder but still effective method is to perturb nodes by a small displacement and then re-evaluate the Jacobian at the new location [Ken04].

5.5 The Colour Map Control

A popular method to visualize a scalar field over a one-dimensional or two-dimensional domain is colour mapping (see subsection 4.6.1). The technique associates a range
5.5 The Colour Map Control

Figure 5.11. The visualization window in the top-middle of the figure shows a zoomed view of the heart model and the field probe with the material coordinate axes being displayed. The white lines indicate the finite elements of the model (see subsection 5.8.1). The top-right window is used to create the field probe and allows the user to define the information which is displayed in the “field probe output” window shown in the bottom-right of the figure.
of scalar field values with a colour spectrum and uses this mapping to represent user-defined subsets of the scalar field in the appropriate colours. Colour maps can also be used to map scalar information onto other visualization icons such as streamlines and streamtubes.

The visualization toolkit contains a global list of colour maps which can be used for multiple icons of the same or different models. A colour map consists of a colour spectrum, a range of field values associated with the spectrum, and a default min- and max-colour indicating values above or below the specified range, respectively.

5.5.1 Colour Spectra

A selection of the colour spectra implemented is shown in figure 5.12 (a). The specifications of the colour scales proposed by Levkowitz and Hermann [LH92] were obtained from [Lev97]. Spectra with hue variations only, such as the rainbow colour scale, are best suited for illuminated surfaces since the surface shading variations do interfere with the brightness variations of a colour spectrum. Spectra with brightness variation only, such as the linear gray scale, are best suited for visualizations employing a large number of different visualization icons since that way interference between icon colours is minimized. Finally colour spectra with hue and brightness variations maximize the number of perceivable different field values.

In some applications it is preferable to define new colour maps, for example to minimize color clashes with existing icons or where specific colours are associated with material properties. Part (b) of figure 5.12 shows the graphical user interface for creating a new color spectrum as a linear interpolation of a set of user defined colors.

Figure 5.13 shows two examples. The spectrum in part (a) of the figure interpolates linearly between saturated yellow, saturated red, and saturated blue and is popular because of its intuitive colour choice: Blue is usually associated with cold temperatures (ice) and is therefore well suited to indicate low field values whereas yellow is associated with high temperatures (flame) and therefore implies high field values. Though the colour scale is popular we are not aware of an official name for it and we term it the temperature scale. We found that the red colour tends to dominate the spectrum which makes the interpretation of the visualization results difficult. An improved version shown in part (b) of the figure is obtained by extending the blue color to black-blue and the yellow colour to white-yellow. The temperature spectra proved useful in our research because they are intuitive, because our collaborators use similar spectra in their research software, and because the spectra use only three hues which leaves the remaining hues for colouring simultaneously displayed icons. A further advantage is that both spectra are relatively robust against simultaneous contrast (caused by neighbouring complementary colours) since they have neither red-green nor yellow-blue boundaries. The spectra could be further improved by linearizing them according to perceived colour differences.

Another tool for creating new colour maps has been implemented in a student project supervised by us [Qu02] and is contained in an extended version of this
5.5 The Colour Map Control

Figure 5.12. (a) Subset of the colour spectra available in the toolkit. (b) User interface for creating piecewise linear colour scales.

Figure 5.13. Two versions of the piecewise linear temperature colour spectrum.
PhD work. Colour maps are defined by constructing a spline curve in a 3D colour space (RGB, HIV, CIE, CIE Lab, CIE Luv). The spline curve is then parameterized with its arclength [HL92b] and sampled at regular intervals resulting in a 1D texture map usable with our colour mapping algorithm. Using a perceptually uniform colour space such as the CIE Lab space results in a perceptually uniform colour map.

If a colour map is changed all visualization icons using this map are marked as changed and can be updated with a single button press. The automatic update proves useful when adding a new model to an existing visualization. For example, if a visualization has been created for the model of the healthy left ventricle and the user wants to compare the results with the sick left ventricle it is sufficient to change the range of all colour maps to reflect the range of field values for both models.

5.5.2 Texture Mapping

An object can be colour mapped either by vertex colouring or by texture mapping. Most visualization icons available in our toolkit, such as colour mapped surfaces, give the user a choice between these two options.

Defining colour mapped objects by specifying vertex colours results in the graphics hardware using bilinear interpolation (Gouraud shading [FvFH92]) to render the polygons. Since the spectrum colours in general do not vary linearly this can result in shading artifacts or, worse, in sections of the colour spectrum being omitted. Figure 5.14 (a) shows that this problem becomes especially obvious when using cyclical colour maps which are explained in the following subsection.

The problems can be alleviated by subdividing the colour mapped surface into more polygons. This, however, is inefficient and memory expensive. Instead we define a colour map as a one-dimensional texture map. For each polygon vertex we define a texture coordinate and render the polygons using texture mapping hardware. We found that 1024 texels, each consisting of three floats for the RGB colour values, are sufficient for all our applications. Using an OpenGL graphics card the rendering times using Gouraud shading and texture mapping do not noticeably vary. The improved graphical representation obtained with texture mapping is illustrated in figure 5.14 (b).

Care must be taken when deciding whether a colour mapped object should be illuminated or not. On the one hand lighting is important for perceiving the 3D geometry of an object. On the other hand shading a coloured surface changes its perceived colour. We illuminate a colour mapped surface if the surface shape encodes important information. An example are hyperstreamlines which are explained in subsection 5.9.2. The use of texture mapping for colour mapping does not impede surface illumination: Texture mapped polygons can be lit in OpenGL by specifying their colour as white, by shading them and by modulating the polygon colour with the texture colour.
5.5.3 Features

We have implemented several features for colour maps in order to increase their effectiveness.

Spectrum Markers

*Spectrum markers* can be inserted into the colour spectrum by specifying their position within the value range of the colour map. A colour map marker appears as isocontours on a colour mapped surface as illustrated figure 5.15 (a). Markers are useful for emphasizing significant values. The example in figure 5.15 (a) shows the 0-isocontour of the principal strain which separates the surface into compressive and expanding regions.

Discrete Colour Maps

Discretizing a colour map into equally spaced constant coloured segments facilitates the comparison of field values in different regions. Also note that the boundaries between the discrete colour regions represent isocontours of the visualized field and the width of a constant coloured region indicates the variation of field values in that region. Discrete colour maps are therefore useful for understanding the structure of the visualized field. An example is shown in figure 5.15 (b).
Figure 5.15. Features of colour maps: (a) spectrum marker, (b) discrete colour map, (c) cyclical colour map, (d) exponential colour map.
Cyclical Colour Maps

Cyclical colour maps map several cycles of a colour spectrum over the specified mapping range. We found cyclical colour maps are especially useful when trying to understand the fine structure of a scalar field and to uncover symmetries and discontinuities [WL99, WL01b]. An example is given in figure 5.15 (c). The image shows clearly some discontinuities of the visualized scalar field along the element boundaries. Also note that the contour density and contour normal direction of a surface mapped with a cyclical colour map indicates the magnitude and direction, respectively, of the visualized scalar field. Furthermore cyclical colour maps can be used to discover symmetries in a data set (e.g., the symmetry between the left and right hemisphere of the human brain).

Exponential Colour Maps

For some applications scalar fields can have large variations in values. An example from the field of structural mechanics is crack propagation, where small areas of extreme stress values occur resulting in visualizations with large constant coloured regions.

We “stretch” a colour spectrum by associating the range of visualized values with a variable $\gamma$ ($0 \leq \gamma \leq 1$) and by replacing $\gamma$ with $\gamma^e$ where $e$ is a user defined exponent. The resulting spectrum is suitable for applications where the minimum of the spectrum range represents extreme field values. If the extreme values are given by the maximum values of the spectrum range the user can invert the spectrum which replaces $\gamma^e$ with $1 - (1 - \gamma)^e$.

The example in figure 5.15 (d) uses a colour map which is stretched so that red and orange represent values close to zero, blue represents large negative values and yellow represents large positive values. While the exponential colour map does not give an exact differentiation of the field into positive and negative regions as is achieved by the colour marker used in part (a) of the figure we believe that the visualization is more intuitive and the structure is more easily perceived than in (a).

5.5.4 Implementation

A colour map consists of a colour spectrum and a spectrum marker. Furthermore it contains a range of scalar values over which the spectrum is mapped, two default colours for values above and below the specified range (by default set to light gray and dark gray, respectively), the number of discretization steps, the exponent and the number of colour cycles. A spectrum marker is specified by an isoalue, a half-width, and a colour.

A colour spectrum is defined over the unit range $[0, 1]$. The colour associated with a given field value $f$ is obtained by computing the value $\gamma \in [0, 1]$ which corresponds to the relative position of the field value with respect to the range $[f_{\text{min}}, f_{\text{max}}]$ of the
colour map, i.e.,
\[ \gamma = \frac{f - f_{\text{min}}}{f_{\text{max}} - f_{\text{min}}} \]

The spectrum colour at that point is then obtained by the virtual method \texttt{indexColourMap()}. The colour map is reversed by setting \( \gamma := 1 - \gamma \). An exponential colour map is created by using \( \gamma := \gamma e^e \) where \( e \) is the selected exponent and for an inverted colour map we set \( \gamma := 1 - (1 - \gamma)^e \). Figure 5.16 shows an example of different colour maps obtained by reversing, inverting and exponentially stretching the temperature scale.

![Figure 5.16. Color maps generated from the temperature scale by reversing, inverting and exponentially stretching it: (a) the original temperature scale (b) the original scale reversed (c) the original scale exponentially stretched (d) the original scale exponentially stretched and reversed (e) the original scale exponentially stretched and inverted (f) the original scale exponentially stretched, reversed and inverted.](image)

Using this design a new colour spectrum is easily defined using inheritance and by overriding the method \texttt{indexColourMap()}. Note that a colour spectrum is a subclass of a texture map and the colour spectrum is stored as a 1024 texel 1D texture map. Figure 5.17 shows a diagram of the colour map class structure.

### 5.6 Volume, Surface and Point Selection

A data set is visualized by creating visualization icons which represent the data at selected points or over surfaces or volumes of interest. Rather than requiring the user to specify the spatial domain of each new icon the toolkit keeps a list of volumes, surfaces and point sets which represent regions-of-interest.

For example, when examining the left ventricle (figure 3.11) medical specialists are particularly interested in its outside surface (epicardial surface), the inside surface (endocardial surface) and the surface in the middle of the heart wall. Using our selection tools the user can define these surfaces and reuse them with different visualization icons such as colour mapped surfaces and line integral convolution textures. Volumes of interest are also an essential component of the “surface sculpturing” tool introduced in subsubsection 5.9.1 which is used to “trim” isosurfaces. For example, in chapter 7 anatomical structures are defined by isosurfaces and artifacts are removed by specifying appropriate regions-of-interest.
Figure 5.17. Diagram of the colour map class structure.

All region-of-interest objects belong to a visualization object and can be created, modified, loaded and saved using the tabs at the bottom of the visualization control window (see figure 5.1).

5.6.1 Volume Selection

Volumes-of-interest can be specified as geometric structures in world coordinates or as sets of elements. A simplified class diagram of the data structure used to implement volume-of-interest selection is shown in figure 5.18.

A volume-of-interest in world coordinates is either a cuboid or a sphere. The user can specify the dimensions of these objects by using a text interface or by modifying the object interactively in the visualization window. The default values of these volumes are given by the dimensions of the bounding box of the underlying finite element model. More complicated regions-of-interest represented by CSG-objects might also be useful and will be implemented in future versions of this toolkit. Figure 5.19 shows the user interface for selecting a volume-of-interest in
world coordinates and figure 5.20 shows two examples of such volumes.

An element set can be specified as a range of element indices, a list of element indices, as elements enclosed by a bounding box, and as elements intersected by a plane. A range of elements proves useful for applications where most or all of the elements of the model are selected. The element list and bounding box are usually employed for selecting a small region of interest. The set of elements intersected by a plane proves useful for the definition of colour mapped surfaces and height fields.

The user interface for selecting element sets is shown in figure 5.21 and examples of the resulting sets are illustrated in figure 5.22. All element set selections implement methods to loop through the set of elements as indicated by the class diagram in figure 5.18.

5.6.2 Surface Selection

A simplified class diagram of the data structure for selecting a surface-of-interest is shown in figure 5.23. Surfaces can be specified in material or in world coordinates.

A plane $\xi_i = c$, $(i = 1, \ldots, 3)$ parallel to a material coordinate plane is characterized by the isovalue $c$ and the index $i$ of the fixed coordinate $\xi_i$. If using curvilinear elements the plane will be a curved surface in world coordinates. The user must specify an element set and only surface patches within these elements are created.
A plane in world coordinates (more precisely a parallelogram) can be defined interactively by the user by positioning three points in the 3D world space. A more general curved surface is created by positioning the control points of a NURBS surface [Far95]. It is also possible to specify a plane \( x_i = c, (i = 1, \ldots, 3) \) parallel to a world coordinate plane by the isovalue \( c \) and the index \( i \) of the fixed coordinate \( x_i \).

The surfaces defined using the above described mechanism can be used to create visualization icons such as colour mapped surfaces. Furthermore the surface selection mechanism is used for the point selection mechanism explained in the following subsection.

Figures 5.24 and 5.25, respectively, show the user interface for creating surfaces-of-interest and some examples of the resulting surfaces. All surfaces in the figure are colour mapped with the major principal myocardial strain. Part (a) of the figure shows from left to right a material coordinate plane \( (\xi_3 = 0) \) selected for all elements (using the input shown in figure 5.24), a plane orthogonal to the world coordinate z-axis, and an interactively specified plane in world coordinates. Part (b) of the figure shows on the left an interactively defined NURBS surface and on the right the same surface colour mapped.
Figure 5.21. User interface for specifying element sets.

Figure 5.22. Examples of element sets.

Figure 5.23. Simplified class diagram of the data structure for selecting a surface-of-interest.
5.6 Volume, Surface and Point Selection

5.6.3 Point Selection

In many applications it is desirable to explore a field by distributing a number of visualization icons uniformly or randomly over a region-of-interest. Typical examples are vector arrows distributed over a flow field or a planar set of seed points for tracing a streamline bundle. Depending on the application either a volumetric point set or a surface-based point set can be more desirable. We have implemented different types of sample point sets which can be used to place visualization icons. The simplified class diagram of the resulting data structure is shown in figure 5.26.

Volumetric point sets include regular grids of points or random point sets in world space over a volume-of-interest and regular grids of points or random point sets in material space over a selection of elements. It is also possible to directly specify a list of points in world or material coordinates. Since a uniformly distributed set of random points in material space is usually not uniformly distributed in world space we introduce additionally a volume-weighted random selection of points where the number of random points for each element is proportional to the element’s volume.

Points specified in material coordinates are usually computationally more efficient since they can be used immediately as parameters of the field interpolation functions (if the field uses a FE interpolation). In contrast, a world coordinate point must be transformed into material coordinates first using a multi-dimensional Newton method.

**Figure 5.24.** User interface for specifying surfaces-of-interest.

**Figure 5.25.** Examples of various surfaces-of-interest.
Figure 5.26. Simplified class diagram of the data structure for creating a point set over a region of interest.
5.6 Volume, Surface and Point Selection

Using sample points defined in material coordinates may also result in a more informative visualization since the material space often reflects the inherent structure of the modelled object (e.g., the anatomic structure of the ventricle). As mentioned above a disadvantage of defining sample points in material space is that the sample density in world coordinates varies according to the volume of an element. Note, however, that this effect is sometimes desirable since small elements are frequently used in finite element modelling to represent regions with large field variations which are of particular interest to the user.

The described point selection mechanisms can be enhanced by defining a boolean filter. Filters are boolean expressions defined over selected data fields and are explained in more detail in the next section. If a filter is specified only points at coordinates for which the boolean expression is true are chosen. This feature can be used to reveal inside structures in a dataset and to prevent visual cluttering.

Figure 5.27. User interface for creating a volumetric point set.

Figure 5.28. Six examples of volumetric point sets.

Figure 5.27 shows the user interface for creating sample point sets. The active tab shown in the figure is used to create a regular grid of sample points in material coordinates within an element set (see subsection 5.6.1).

Some examples of volumetric point sets are illustrated in figure 5.28. The two images in the top row of the illustration depict a regular grid in material coordinates over 5 of the 10 elements of the model and a regular grid in world coordinates over half of the bounding box of the model. The middle row shows on the left a random point set in material coordinates (using all elements) and on the right a random point set in world coordinates within a spherical region-of-interest. The left image in the bottom row illustrates a volume-weighted random point set in material coordinates.
Note that in contrast to the image above it large elements contain more points than small elements resulting in a more even distribution in world coordinates. Finally the image in the bottom-right of the figure shows an example of applying boolean filters, introduced in the next subsection, to a volume weighted random point set. A filter implements a boolean expression containing data fields. Applying it to a point set has the effect that all points for which the boolean expression is false (i.e., the fields at that point do not fulfill the specified conditions) are removed.

*Surface-based point sets* include spiral point sets and regularly or randomly sampled surfaces-of-interest defined with the previously described surface selection tool. Figure 5.29 shows the user interface for creating sample point sets over surfaces. The active tab in this illustration is that for creating a regularly sampled surface.

A surface is sampled by dividing its parameter space into equally spaced steps. The parameter space of a NURBS surface is defined by its knot vectors and the parameters of a plane in material space ($\xi_i = c$) are $\xi_j$ ($j = 1, ..., 3; j \neq i$). A quadrilateral uses the parameters of the bilinear interpolation of its vertex coordinates. Similarly a surface defined as a plane in world coordinate space ($x_i = c$) uses the parameters of the bilinear interpolation of the coordinates of the rectangle formed by intersection the model’s bounding box with the coordinate plane.

Finally a spiral point set of size $N$, distributed over the surface of a sphere, is defined by [RSZ94]

$$\theta_k = \cos^{-1}h_k, \quad h_k = -1 + \frac{2(k-1)}{(N-1)}, \quad 1 \leq k \leq N$$

$$\phi_k = \left(\phi_{k-1} + \frac{3.6}{\sqrt{N} \sqrt{1-h_k^2}}\right) \pmod{2\pi}$$
where \((\theta, \phi), \; 0 \leq \theta \leq \pi, \; 0 \leq \phi \leq 2\pi\) are the spherical coordinates of the points. The world coordinates of the resulting points are

\[
(x_k, y_k, z_k) = (r \cos \phi \sin \theta, r \sin \phi \sin \theta, r \cos \theta)
\]

where \(r\) is the radius of the sphere over which the points are distributed. Saff and Kuijlaars report that this construction has for large \(N\) considerable advantages over other algorithms for distributing points over a sphere \([SK97a]\).

Figure 5.30 shows from top to bottom a regularly sampled NURBS surface, a randomly sampled surface in material coordinates, and a spiral point set. Only points lying within the model are shown.

Points regularly distributed over a surface in world coordinates are useful as seed points for streamlines since the changes in distance between neighbouring streamlines indicate divergence and convergence in the field. Spiral point sets can also be used to generate seed points for streamlines and are ideal for examining topological flow features such as sources in a flow field from where streamlines emanate in all directions.

5.7 Filters

In many instances visualization icons are only required in regions with interesting field properties. Such regions can be specified by using filters which are boolean expressions defined over the model domain containing three types of terms: a comparison between scalar fields or constants, e.g., “0.5 < density(x)”, a range expression, e.g., “0 < density(x) < 1”, or a probabilistic expression where the probability that the expression is true is determined by the value of a scalar field. The maximum value of the scalar field gives a probability of one and the minimum value a probability of zero. New filters can be constructed from existing filters by combining boolean expressions using the logical operations AND, OR, XOR, NOT, and IMPLIES.

The toolkit uses filters for three tasks: The first task is the definition of point selections. Any of the previously introduced point selection mechanisms can be supplemented with a filter. A sample point is selected only if the boolean expression at that point is true. This tool is useful for the creation of visualization icons at regions of interest, e.g., points where the blood flow velocity exceeds a certain limit. Figure 5.31 gives an example.

The second task for which filters are used is the definition of conditional expressions in the field data structure introduced in section 5.3. All conditions of such an expression are represented by filter objects. An interesting application of this tool is given in chapter 7 where we use conditional expressions for the visual segmentation of a brain data set by tissue types. The user can interactively adjust values to improve the segmentation result.

Finally filters are also useful for specifying the shape of a visualization icon. For example, when defining streamlines and streamtubes filters can be used as an integration condition. The integration along a vector field is continued as long as
the filter condition is true along the trajectory of the streamline. Chapter 7 will explain how this tool can be used to extract the nerve fiber structure in the brain from a diffusion tensor data set [Wën02].

5.8 Model Geometry

Visualizing the geometry of a finite element model facilitates the understanding of the 3D orientation and position of scene components with respect to the underlying structure. The FE model can also be used to compute properties of the biomedical structure it represents. An example is given in chapter 6 where the left ventricular FE model introduced in subsection 3.2.8 is used to compute important ventricular output measures.

5.8.1 Visualizing the FE Model Geometry

The display of the FE model geometry is important since it improves the perception of a visualized data set. In the simplest case where the underlying FE model is the bounding box of the data set the model boundary serves as a scale. Comparing the position of a point to the edges of the bounding box helps estimating its 3D coordinates and makes it easier to compare the distances between scene components. The model outline also improves the perception of the 3D geometry and orientation of features within a visualized structure.

The model geometry can be displayed either as a wireframe or by rendering element surfaces. Figure 5.32 shows on the left the user interface for creating a wireframe representation. In general a wireframe should be displayed with a colour which improves its perception and is clearly distinguishable from other scene com-
ponents. Alternatively the user can colour map the wireframe with a scalar field. Dependent on the resolution of the FE model this feature can yield an excellent understanding of the 3D distribution of the visualized field. Also note that for many FE simulations the extreme values are at or near the nodes of the FE mesh so that colour mapping the mesh can reduce the risk of missing important features in the visualized data set. Finally, a colour mapped wireframe uses the screen space effectively and therefore helps to reduce visual cluttering.

A complex model geometry is often difficult to perceive if rendered as a wireframe. Perception is improved by using shaded element surfaces instead. Often it is possible to select a subset of element surfaces which illustrates the object geometry but also allows a view inside the model and its visualization. Element surfaces are also useful for reducing visual cluttering (by hiding complex scene components in the background) and for producing anatomical structures such as the ventricular cavities (see chapter 6).

The user interface for creating the model surface is shown in figure 5.33 on the left. Pressing the button “Set Model Surface” automatically selects all element faces which are part of the boundary surface of the FE model (i.e., faces which are not shared by two elements). The selection can be modified by pressing the button “Change Surface” which brings up the user interface shown in the middle of the figure. The user can add or delete either individual element faces or the faces of a set of elements specified using the element selection mechanism explained in subsection 5.6.1. The image on the right of figure 5.33 shows the surface of a FE model.

In some instances it is preferable to visualize the surface of the FE model together
with other objects on the surface. An example is the simultaneous display of the boundary surface and the wireframe of a FE model as demonstrated in figure 5.33. If two objects with the same coordinates are rendered with OpenGL it is undefined which one is visible. In practice, however, it is usually desirable to show all objects lying on the object surface. This effect is obtained by selecting the check box “Enable polygon offset” which adds a small amount to the depth buffer values of the surface polygons using the OpenGL commands

```c
glEnable(GL_POLYGON_OFFSET_FILL);
glPolygonOffset(0.2f, -0.4f);
```

### 5.8.2 Computing Model Properties from the FE Geometry

The health of an anatomical structure such as the heart is often quantified using various volume, area and length measures. For example, the performance of the left ventricle is described by its systolic and diastolic volume and its ejection fraction. Our visualization toolkit enables the user to specify elements, parameter surfaces and parameter curves of a FE model and to compute their volume, area and length, respectively.

#### Computing Volume Measures

The volume of a single element is obtained by integrating the identity function over the finite element in world coordinates. The calculation is simplified by using the substitution rule of multi-dimensional integration [Heu81, p.478] which gives

\[
\int_{\Omega} 1 \, du = \int_{\Omega} |det \, J(\xi)| \, d\xi \tag{5.1}
\]
5.8 Model Geometry

where $\Omega$ is the unit cube representing the domain of the parent element, $x(\xi)$ is the transformation function from $\xi$-coordinates to world coordinates and $J$ is its Jacobian. The resulting integral can be evaluated efficiently using Gaussian Quadrature (see subsection 2.4.4).

**Computing Surface Measures**

The area of a surface $\Phi = \Phi(u,v)$ over a parameter region $K$ is computed by [Heu81, p.505]

$$I(\Phi) = \int_K \left| \frac{\partial \Phi}{\partial u} \times \frac{\partial \Phi}{\partial v} \right| d(u,v) = 
\int_K \sqrt{ \left( \frac{\partial x_1}{\partial u} \right)^2 + \left( \frac{\partial x_2}{\partial u} \right)^2 + \left( \frac{\partial x_3}{\partial u} \right)^2 + \left( \frac{\partial x_1}{\partial v} \right)^2 + \left( \frac{\partial x_2}{\partial v} \right)^2 + \left( \frac{\partial x_3}{\partial v} \right)^2 } d(u,v) \tag{5.2}$$

We are mostly interested in surfaces parallel to one of the material coordinate axes. For example, the endocardial surface of the left ventricle is given by the $\xi_3 = 0$. In this case $\Phi(\xi_1, \xi_2) = x(f(\xi_1, \xi_2))$ where $f(\xi_1, \xi_2) = (\xi_1, \xi_2, 0)$ and

$$\frac{\partial \Phi}{\partial \xi_1} = \frac{\partial x}{\partial f_1} \frac{\partial f_1}{\partial \xi_1} + \frac{\partial x}{\partial f_2} \frac{\partial f_2}{\partial \xi_1} + \frac{\partial x}{\partial f_3} \frac{\partial f_3}{\partial \xi_1} = \frac{\partial x}{\partial \xi_1} \quad \text{and similarly for the other partial derivatives.}$$

The surface area $A$ is therefore given by

$$A = I(\Phi) = \int_0^1 \int_0^1 \sqrt{ \left( \frac{\partial x_1}{\partial \xi_1} \right)^2 + \left( \frac{\partial x_2}{\partial \xi_1} \right)^2 + \left( \frac{\partial x_3}{\partial \xi_1} \right)^2 + \left( \frac{\partial x_1}{\partial \xi_2} \right)^2 + \left( \frac{\partial x_2}{\partial \xi_2} \right)^2 + \left( \frac{\partial x_3}{\partial \xi_2} \right)^2 } d\xi_1 d\xi_2$$

where the partial derivatives $\frac{\partial x_j}{\partial \xi_i}$ are the elements of the Jacobian $J$ of the coordinate transformation function $x(\xi)$.

**Computing Length Measures**

Similar to the volume and area computations it is also possible to compute the arc-length of a parametric function $\gamma : [a, b] \to \mathbb{R}^3$ [Heu81, p.354]

$$L(\gamma) = \int_a^b |\gamma(t)| dt = \int_a^b \sqrt{\gamma_1^2 + \gamma_2^2 + \gamma_3^2} dt \tag{5.3}$$

Assume the start point and end point of a parameter curve within a finite element are $\xi_s$ and $\xi_e$. Then $\gamma(t) = x(\xi(t))$ where the line segment $\xi(t) = \xi_s + t(\xi_e - \xi_s)$ with $t \in [0,1] = [a, b]$ is the curve in material coordinates. Hence

$$\gamma(t) = \frac{\partial x}{\partial \xi} \frac{\partial \xi}{\partial t} = J(\xi(t))(\xi_e - \xi_s)$$

where $J$ is again the Jacobian of the transformation function from $\xi$-coordinates to world coordinates.
Numerical Integration

In general the element integrals in equations 5.1-5.3 can not be computed analytically and a numerical integration is necessary. Since the basis functions of the finite element method are polynomial the integration method of choice is Gaussian quadrature (see subsection 2.4.4).

We compare Gaussian integration with alternative integration algorithms using as an example the computation of the myocardial volume of the left ventricle at end-diastole and end-systole (see figures 3.11-3.12).

As explained in subsection 3.2.8 the left ventricle is modeled using bicubic-linear interpolation functions $\Psi_i^k$ which are constructed analogously to the bicubic interpolation functions introduced in subsection 2.4.2. In order to choose the correct number of gauss points consider the Jacobian $J = \left(\frac{\partial x_i}{\partial \xi_j}\right)$ of the finite element transformation function

$$x(\xi) = \sum_i \left( \Psi_i^1 \mathbf{u}_i + \Psi_i^1 \left( \frac{\partial \mathbf{u}}{\partial \xi_1} \right)_i + \Psi_i^2 \left( \frac{\partial \mathbf{u}}{\partial \xi_2} \right)_i + \Psi_i^{12} \left( \frac{\partial \mathbf{u}}{\partial \xi_1 \partial \xi_2} \right)_i \right)$$

which interpolates the nodal coordinates and coordinate derivatives with the element basis functions. Since the basis functions $\Psi_i^k$ are bicubic-linear the highest degree term of $x(\xi)$ is $\xi_1^3 \xi_2^3 \xi_3$ so that the highest degree term in $\text{det} J(\xi)$ is $\xi_1^8 \xi_2^8 \xi_3^2$. Assuming that the determinant of the Jacobian has a constant sign, which is the case for non-degenerate finite elements, the integral in equation 5.1 is evaluated exactly by choosing 5 gauss points in the $\xi_1$ and $\xi_2$ directions and 2 gauss points in the $\xi_3$ direction.

In order to test the suitability of different Gaussian quadrature formulas we use them to compute the left ventricular volume and compare the results with alternative integration methods. The first integral approximation divides the material space into $n^3$ cubes and approximates the corresponding regions in world space by parallelepipeds whose volume is computed using the scalar-triple product [Wei]. The second method is a multi-dimensional Newton-Cotes integration with $n$ steps in each dimension [eFu] and the third method is a Monte-Carlo integration with $n^3$ sample points [Wei]. We compare these methods for $n=10$, $n=50$, and $n=100$ with different Gaussian quadrature formulas.

Table 5.1 demonstrates that Gaussian quadrature is superior to all alternative integration methods tested. As explained above the correct integral is obtained by choosing 5x5x2 Gauss points. Choosing 3x3x2 Gauss points gives 4 figure accuracy and using 4x4x2 Gauss points achieves (at least) 6 figure accuracy. Since Gaussian quadrature is also considerably faster than the other examined techniques we use it for all numerically evaluated FE integrals.

User Interface

Figure 5.34 shows the user interface for computing the volume of an element set (left), the user interface for computing the area of a surface (middle), and the user interface for computing the arc length of a parameter curve (right).
5.8 Model Geometry

| Approximation by parallelepipeds ($n = 10^3$) | 203206 | -6.60 | 146543 | -7.90 |
| Approximation by parallelepipeds ($n = 50^3$) | 214313 | -1.50 | 156460 | -1.67 |
| Approximation by parallelepipeds ($n = 100^3$) | 215493 | -0.96 | 157540 | -0.99 |
| Numerical Integration (Newton-Cotes $n = 10^3$) | 229851 | 5.64  | 166543 | 4.04  |
| Numerical Integration (Newton-Cotes $n = 50^3$) | 219954 | 1.09  | 160357 | 0.78  |
| Numerical Integration (Newton-Cotes $n = 100^3$) | 218760 | 0.54  | 159732 | 0.39  |
| Numerical Integration (Monte-Carlo $n = 10^3$) | 218521 | 0.43  | 159906 | 0.50  |
| Numerical Integration (Monte-Carlo $n = 50^3$) | 217606 | 0.01  | 159136 | 0.02  |
| Numerical Integration (Monte-Carlo $n = 100^3$) | 217602 | 0.01  | 159135 | 0.01  |
| Numerical Integration (3*3*2 Gauss Points) | 217594 | 0.01  | 159138 | 0.02  |
| Numerical Integration (4*4*2 Gauss Points) | 217575 | 0.00  | 159112 | 0.00  |
| Numerical Integration (5*5*2 Gauss Points) | 217575 | 0.00  | 159112 | 0.00  |

<table>
<thead>
<tr>
<th>End-diastole</th>
<th>End-systole</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume (in mm$^3$)</td>
<td>Error (in %)</td>
</tr>
<tr>
<td>203206</td>
<td>-6.60</td>
</tr>
<tr>
<td>214313</td>
<td>-1.50</td>
</tr>
<tr>
<td>215493</td>
<td>-0.96</td>
</tr>
<tr>
<td>229851</td>
<td>5.64</td>
</tr>
<tr>
<td>219954</td>
<td>1.09</td>
</tr>
<tr>
<td>218760</td>
<td>0.54</td>
</tr>
<tr>
<td>218521</td>
<td>0.43</td>
</tr>
<tr>
<td>217606</td>
<td>0.01</td>
</tr>
<tr>
<td>217602</td>
<td>0.01</td>
</tr>
<tr>
<td>217594</td>
<td>0.01</td>
</tr>
<tr>
<td>217575</td>
<td>0.00</td>
</tr>
<tr>
<td>217575</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 5.1. Myocardial volume of the healthy left ventricle at end-diastole and end-systole computed using different integration methods.

Currently the area computation is only implemented for material coordinate planes which are characterized by having a constant $\xi_i$ parameter. This type of surface defines the boundaries of finite element models and is hence often of interest in practice. For example, the $\xi_3 = 0$ and $\xi_3 = 1$ surface of the left ventricular model represent the endocardial and epicardial surface, respectively. In some applications the area and enclosed volume of an isosurface might be important and appropriate computational methods might be added in future.

The user interface for computing length measures is currently restricted to the arc length of parameter curves which are defined by fixing two material coordinates of a finite element and varying the remaining one. Future versions of this toolkit might be extended with a selection tool which allows the specification of a wider variety of curve representations within the model.

Figure 5.34. The user interfaces for computing the volume of an element selection (left), the area of a surface (middle) and the arc length of a parameter curve (right).
5.9 Visualization Icons

So far this chapter has presented the overall structure of our visualization environment and some general tools. This section introduces the currently implemented visualization icons and explains their effective usage. The visualization icons can be created, modified, saved and loaded using the visualization control pictured on the left of figure 5.1.

![Figure 5.35. A bulged plate with a hole (a) and the corresponding model of 1/4 of the plate under an uniaxial load \( F \).](image)

The application of the visualization icons is demonstrated with a popular problem in linear elasticity: the stress and strain in a plate with a hole under an uniaxial load. Figure 5.35 (a) shows a plate under an uniaxial load \( F \) in \( x \) direction. To make the resulting strain and stress field more interesting the plate is thickened (bulged) around the hole. Because of symmetry only one quarter of the plate is modelled (b). In order to get a unique solution, boundary conditions must be specified. We fix the \( x \)-coordinate of the nodes on the left face and the \( y \)-coordinate of the nodes of the front face of the plate since these faces will not move in the corresponding coordinate directions because of the symmetry of the model. We fix additionally the \( z \)-coordinate of the nodes on the bottom right edge in order to get a unique solution. The data set was computed using the formulas derived in appendix D.2.

Since visualization icons are explained in chapter 4 this section concentrates on the graphical user interfaces for defining and placing visualization icons, the application of icons to a visualization problem, and the improvements and changes in our implementation over those described in the literature.

Please note that the visualizations shown in this section illustrate the features of the toolkit and do not attempt to create an optimal visualization of the stress and strain field in the plate with a hole. In fact, some examples will demonstrate that the careless use of certain icons does indeed obscure information and leads to poor visualizations.

5.9.1 Scalar Icons

Colour Mapped Surface

Colour mapped surfaces are ideal to visualize the distribution and structure of a scalar field over a 2D domain. The user interface for creating a colour mapped surface is shown in the left part of figure 5.36. Any surface-of-interest defined with
the surface selection tool introduced in subsection 5.6.2 can be colour mapped. The user can choose whether the surface is texture mapped or Gouraud shading (see section 5.5).

In order to achieve a smooth surface representation each surface patch is approximated by \( n^2 \) polygons where \( n \) is selected using the “Number of refinements” slider. A NURBS surface or a surface in world coordinates consists of only one patch whereas a surface defined in material coordinates has one surface patch for each element.

The user can also select the visualized scalar field and the colour map. In order to make it easier to select or define an appropriate colour map the user interface displays the currently selected colour map and the range of the currently selected scalar field. The middle part of figure 5.36 shows two examples of colour mapped surfaces. The visualization shows clearly that extreme stress values occur at the left bottom corner of the hole (in yellow) but no conclusions can be drawn about the stress distribution inside the plate.

![Figure 5.36](image)

**Figure 5.36.** The user interface for creating a colour mapped surface (left), two colour mapped surfaces used to visualize the maximum principal stress in the plate with a hole (middle) and the colour mapped surface of the plate model with transparencies enabled (right).

If a colour mapped surface obstructs other icons the user can enable surface transparencies. Figure 5.36 shows on the right a scalar field visualized by colour mapping its boundary surface and making it partially transparent.

Transparencies are currently not fully implemented and are simulated using the OpenGL blending function [WND97]. The colour of a pixel on the screen is computed as \( \alpha C_d + (1 - \alpha) C_s \) where \( \alpha \) is the opaqueness, \( C_d = (R_d, G_d, B_d, T_d) \) is the current pixel colour, and \( C_s = (R_s, G_s, B_s, T_s) \) is the colour of the currently drawn object. The colour components and the transparency value are indicated by \( R, G, B, \) and \( T \). Note that for several overlapping objects the result is dependent on the drawing sequence and not, as it should be, on the depth value of a pixel. This means that blending does not create “real” transparencies. However, an experienced user can
create interesting visualizations with this technique. Correct transparencies could be achieved by depth sorting polygons using a BSP-tree [SBGS69, FKN80].

A disadvantage of using transparencies is that they make it more difficult to perceive actual colour values, i.e., blending reduces the information content of the visual attribute colour which reduces the effectiveness of any visualization icon using this attribute (see table 4.2-4.4).

**Isosurface**

The *c*-isosurface of a scalar field \( s \) is defined as all points \( x \) for which \( s(x) = c \). Isosurfaces impart knowledge about the overall distribution of a scalar field and are best combined with continuous field representations, such as colour mapped surfaces, in order to achieve optimal results.

We have developed a modified Marching Cubes algorithm [LC87] which computes an isosurface in material space. The algorithm divides the cubic parent element of each (potentially curvilinear) finite element into a regular grid of \((n + 1)^3\) sample values which form \( n^3 \) cubes in material space. The algorithm determines how the surface intersects a cube, then moves to the next cube. The isosurface intersection is determined by the sign of the scalar field at the cube’s vertices. Each edge with vertex values of different sign is assumed to intersect the isosurface once. The intersection point is approximated by linearly interpolating the scalar field values between the vertices.

Since there are eight vertices in each cube and two values, *positive* and *negative*, there are \( 2^8 = 256 \) ways the surface can intersect the cube. Lorensen and Cline use symmetries to reduce the number of patterns to 15 which are shown in figure 5.37. Since the complete table of 256 cases is still very small we use it directly without eliminating topologically equivalent configurations.

The algorithm can be summarized as

- Subdivide the material space into cubic cells.
- Calculate an 8-bit index for each cube from the sign of the eight scalar field values at the cube vertices.
- Using the index, look up the list of edges forming triangles from a precalculated table.
- Using the scalar field values at each edge vertex linearly interpolate the isosurface intersection and compute its world coordinates.

The surface normals of the isosurface are given by the field’s gradient function if it is defined and if its use is appropriate. Otherwise the normals are determined by first precomputing the material coordinate gradients for all grid points using finite differences. For each isosurface intersection the \( \xi \)-derivative of the scalar field \( s \) at

\[2\]The cases 12 and 15 are reflective with respect to the xy-plane. This leaves 14 topologically distinct patterns (22 without inversed patterns) [LVG80].
that point is then approximated by linearly interpolating the gradients at the grid vertices. Finally the surface normal is given by the gradient in world coordinates which is

$$\nabla s = \begin{pmatrix} \frac{\partial s}{\partial x_1} \\ \frac{\partial s}{\partial x_2} \\ \frac{\partial s}{\partial x_3} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{3} \frac{\partial s}{\partial \xi_i} \frac{\partial \xi_i}{\partial x_1} \\ \sum_{i=1}^{3} \frac{\partial s}{\partial \xi_i} \frac{\partial \xi_i}{\partial x_2} \\ \sum_{i=1}^{3} \frac{\partial s}{\partial \xi_i} \frac{\partial \xi_i}{\partial x_3} \end{pmatrix} = \nabla \xi s \mathbf{J}^{-1}$$

where $\mathbf{J}^{-1} = \frac{\partial \xi_i}{\partial x_j}$ is the inverse of the Jacobian of the isoparametric mapping (see equation 2.25) and $\nabla \xi s$ is the gradient of $s$ with respect to the material coordinates.

Performing the isosurface computation in material space has the advantage that scalar field values can be computed directly without performing a multi-dimensional Newton method or resampling the data. Also the resulting isosurface lies smoothly inside the finite element, i.e., there are no bits of the isosurface sticking out of the model boundaries and there are no erroneous results due to sample values which lie outside the model boundary and for which the scalar field is undefined. Finally the computation in material space is often more precise. For example, if tricubic elements are used then the linear interpolation used to compute the intersection points of the cube’s edges with the isosurface will yield the exact result. In contrast the computation in world coordinates is exact only if the elements are cuboidal (which is usually not the case as can be seen from the “plate with a hole example”) and if the sample grid is aligned with each element.

It is interesting to note that the modified Marching Cubes algorithm is stable even if degenerate finite elements are used. Figure 5.38 gives an example of an element which has two pairs of vertices with the same world coordinates. If we approximate the finite element with a single MC cell then we obtain two triangles (configuration 9) where one of the triangles is a line since the two edge intersections on the right face have the same world coordinates. Our algorithm automatically
removes such degenerate triangles since they are not rendered anyway and since we
might in future want to implement algorithms which use the polygonized isosurface
as input.

The main disadvantage of the algorithm is that some patterns in figure 5.37
are topologically ambiguous as noted by van Gelder and Wilhelms [vW94, pages
343 – 344]. This may produce a surface with a hole as pointed out by Düurst [Düu88]
(see figure 5.39). Also in some applications the topology of a biomedical structure
is known in advance and specialised polygonization algorithms could be employed
to take this into consideration [BK02]. An optimized version of our algorithm is
presented in [WL03].

The literature offers various solutions to the ambiguity problem [NH91, MSS94,
Che95, LM00, Nie03, LB03]. We have surveyed and analysed polygonization meth-
ods and optimization techniques to achieve faster computation and rendering of
isosurfaces [Wün97]. While we haven’t experienced any problems with the current
implementation we intend to incorporate some of the improved methods in future
versions of this toolkit.

Figure 5.40 shows on the left the user interface for creating an isosurface. Since
the isosurface is constructed in material space the user must select a set of elements
containing the surface (usually all elements are selected).

Figure 5.38. Isurface within a degenerate finite element approximated with one Marching
Cubes cell.

Figure 5.39. A hole in the polygonization because of a face ambiguity.
A “Surface Sculpturing” tool allows the user to select or cut off parts of the isosurface after computation. This is done by defining interactively a region in world space and by specifying whether the part of the surface inside this region is selected or removed. The tool was motivated by our observation that for many medical imaging data sets anatomical structures can be approximated by isosurfaces. Because of image noise and the proximity of unrelated structures such isosurfaces frequently contain unwanted bits which can be eliminated with the “Surface Sculpturing” tool. Chapter 7 demonstrates that this tool can be used to approximate the eyes and the ventricles inside the brain by isosurfaces.

The colour of an isosurface is either specified directly or is given by indexing a colour map with the surface’s isovalue. The latter choice is convenient when combining isosurfaces with other icons using the same colour map. A user-defined colour can help preventing ambiguities and differentiating between multiple icons.

Rendering options include the number of subdivisions \( n \) for each material co-ordinate of a parent element and options for calculating the surface normal of the isosurface. Note that isosurfaces are always illuminated in order to improve the perception of their 3D geometry. While illumination makes it difficult to perceive the correct object colour this is not a problem for isosurfaces since they are always single-coloured and the only quantitative information is represented by the isovalue.

![Figure 5.40. The graphical user interface for creating an isosurface (left), the maximum principal stress visualized using four isosurfaces (middle) and the same visualization after applying the “sculpturing” tool (right).](image)

The results of visualizing the plate with a hole with isosurfaces are shown in figure 5.40. The image in the middle shows the 100-, 200-, 300-, and 400-isosurfaces of the maximum principal stress. The region indicated by the red square is shown
enlarged in the image on the right. The image displays the visualization after applying the “sculpturing” tool using a single cuboid (in yellow) and retaining the isosurfaces within the cuboid.

In contrast to the visualizations shown previously the isosurfaces convey an impression of the overall distribution of the scalar field. For example, it can be seen that the maximum principal stress indeed occurs at or close to the left bottom corner of the hole in the plate.

**Height Field**

Height fields are offset surfaces where the offset to the reference surface at each point is proportional to the value of the visualized scalar field at that point. Since quantitative information is best visualized by length or position along a scale (see subsection 4.4.5) height fields can improve the precise perception of field values.

![Figure 5.41. The user interface for creating a height field (left) and the maximum principal stress on the bottom face of the plate with a hole visualized with a colour mapped height field (right).](image)

The user interface for creating a height field, shown in figure 5.41 on the left, allows the specification of the scalar field which defines the surface offset, the scale factor for the offset, and the scalar field which is colour mapped onto the height field. The reference surface of the height field can be any surface of interest except of NURBS surfaces.

NURBS surfaces are usually not suitable for creating height fields since they have often a large and non-uniform curvature which hampers the perception of height values and can cause self-intersections of the offset surface. The most suitable surface
for creating height fields are planes in world coordinates. However, we found that perception of height values is also satisfactory for planes in material space, such as the epicardial surface of the heart, if they have an approximately uniform curvature.

Figure 5.41 shows on the right an example visualization. The reference surface is the bottom face of the plate with a hole and the surface height and colour encode the maximum principal stress. It can be seen that the height values considerably improve the perception of quantitative and qualitative values. For example, the height field reveals that the maximum principal stress over the bottom of the plate is lowest at the front corner of the hole and its value is approximately constant over the right hand side of the plate.

Particle Systems

A good impression of the structure of a scalar field can be obtained by colour coding and randomly distributing a set particles over the field domain. For example, the image in the middle of figure 5.42 indicates that the maximum principal stress in our example data set occurs at the left bottom corner of the hole.

![Figure 5.42. The user interface for creating a particle system (left), the maximum principal stress visualized using volume-weighted randomly distributed particles (middle) and the regions of positive minimum principal stress (yellow-red) and maximum principal stress (green-blue) visualized using two particle systems with different particle sizes and colour maps (right).](image)

Compared with colour mapped surfaces particle systems have the advantage that the scalar field is visualized over a 3D domain. Since the visualization is not “dense” structures inside the model can often be observed. The 3D effect can be improved by making particles partially transparent. Transparency is implemented by using the OpenGL blending mode explained on page 171, by depth sorting the particles with respect to the view point and by rendering them back to front.

In order to reveal more structures in a field the point set defining the particle
positions can be filtered, i.e., particles are displayed only at locations where the field fulfills a user defined criteria (see section 5.7). An example is shown in figure 5.42 on the right. The image illustrates the regions of positive minimum principal stress (yellow-red) and positive maximum principal stress (green-blue) visualized using two particle systems with different particle sizes. It can be seen that the maximum principal stress is positive everywhere whereas the minimum principal stress is positive around the back side of the hole and in some regions close to the back face of the plate. The maxima of both the minimum and maximum principal stress occur at the left bottom corner of the hole.

The graphical user interface for creating particle systems is shown on the left of figure 5.42. The particle positions are specified using the point selection tool. In general the best visualization results are obtained by using random sampling in world coordinates or volume-weighted random sampling in material coordinates.

Figure 5.43. Isosurfaces can be identified by moving particles along the gradient of a scalar field until they reach the required isovalue.

The user interface allows also the creation of isosurface finding particles as shown in figure 5.43. Isosurfaces are determined by moving particles along the gradient of a scalar field until they encounter the user-defined isovalues. If a particle is caught in a local extremum of the field a new start position is randomly generated. Perception of the isosurface is improved by illuminating the particles. The “surface normal” of a particle is given by the field’s gradient.
5.9 Visualization Icons

Probabilistic Distributions

As a new method to visualize scalar fields we propose to use field values to control the distribution density of a discrete visualization icon over the visualization domain. A higher density reflects a higher value of the underlying scalar field. The method was motivated by experiments with probabilistic filters for point selection (see section 5.7) but might also be useful on its own. Since a high number of icons must be drawn in order to perceive a density value, particle fields are the most suitable objects for this application.

Another application for the probabilistic distribution would be to reflect the level of uncertainty or confidence in the data [Ken04]. Similarly the technique can be used to reflect the degree of interest in different regions of the data in order to reduce visual cluttering. For example, if visualizing a fluid flow it is often desirable to draw more vector icons in regions of high flow velocity or rapid changes in the flow direction whereas tensor icons for the viscous stress are drawn predominantly in regions of high principal viscous stresses.

The probability $\lambda$ to draw an icon at a random point $p$ is given by

$$\lambda = \frac{s(p) - s_{\text{min}}}{s_{\text{max}} - s_{\text{min}}}$$

where $s_{\text{max}}$ and $s_{\text{min}}$ are the maximum and minimum values, respectively, of the scalar field $s$. A particle is drawn if $\lambda > \text{rand}()$ where $\text{rand}()$ returns a random value between zero and one.

The above function gives visually poor results since the likelihood of two randomly drawn particles overlapping on the screen increases with the icon density. As a result of this overlap the perception of the actual icon density is impaired. An improvement is achieved by using an exponential probability function, i.e., an icon is drawn if $\text{rand}() < \lambda^e$ where $0 \leq e \leq 1$.

Figure 5.45 shows a probabilistic particle field visualizing the maximum diffusivity over a horizontal section of the brain. The result is inferior to using a colour mapped surface (figure 5.44), but the same basic structures can be recognized. In contrast to colour mapped surfaces probabilistic particle fields can also be used to visualize volumetric regions and the particles can be colour mapped in order to encode other scalar fields. The comparison of particle densities with particle colours can be used to detect correlations between the visualized fields.

5.9.2 Vector Icons

Vector Glyphs

As explained in section 4.7 vector fields can be either signed or unsigned. Signed vectors are represented with vector arrows and unsigned vectors, such as eigenvectors, with thin cylinders. The user interface for creating a set of vector glyphs is shown in figure 5.46 on the left. The vector glyphs are placed using the point set selection tool described in subsection 5.6.3. Various rendering parameters are available and
are explained in the following paragraphs. For most applications the default values are sufficient.

An example is shown in figure 5.46. The visualization uses regularly distributed vector arrows to illustrate the displacement field in the plate with a hole under an uniaxial load. It can be seen that the displacement on the (fixed) left side of the plate is almost constant in $z$-direction (i.e., the plate moves downward) whereas
closer to the right hand side the displacement is predominantly in the x-direction (the direction of the uniaxial load), i.e., the plate is stretched in this direction.

However, it is difficult to determine how the material deforms in the y-direction. In order to allow a more detailed analysis of vector fields we implemented the projection of vector glyphs onto a surface. Projected vectors enable the user to better recognize directional variations transverse to the normal direction of a surface and can be used to increase the information content of colour mapped surfaces. In general projected vectors are useful for visualizing flow near surfaces, but the loss of one dimension makes them misleading for arbitrary flows [Ken04].

The current implementation checks whether the selected point set is sampled over a surface or a volume and in the former case projects the vectors onto the underlying surface. The image on the right of figure 5.46 shows that the back side of the hole moves towards the hole’s centre axis whereas the right hand side of the hole moves away from the centre axis. The means the circular cross section of the hole deforms into an ellipsoid.

Vector glyphs are by default illuminated since shading gives important shape and depth cues. However, shaded vector glyphs require more screen space and unfavorable lighting conditions make their perception difficult. The user can disable lighting which gives the vector glyph a 2D appearance. This feature is useful if the vectors vary only in two dimensions (e.g., if the vectors are projected onto a surface) and if a high vector glyph density is required.

In many applications unsigned vector fields are associated with scalar fields (e.g., an eigenvector field is associated with an eigenvalue field) and the scalar field’s sign is essential for the correct interpretation of the vector field. The user interface for creating vector glyphs allows the selection of an associated scalar field. The sign of this field is encoded into the glyph either by using different colours or by using bidirectional arrows whose heads point apart or towards each other to indicate positive and negative scalar field values, respectively.

Figure 5.47 gives an example. Both parts of the figure visualize the displacement vector field with yellow arrows. Part (a) of the figure indicates the direction of the minimum principal stress with bidirectional arrows and part (b) of the figure uses red and blue coloured cylinders to indicate positive and negative values for the minimum principal stress, respectively. Both parts of the figure suffer from a high degree of visual cluttering which indicates that it is problematic to visualize two different vector fields at once with vector glyphs. In contrast to part (a) of the figure it can be seen immediately from part (b) of the figure that the minimum principal stress is compressive everywhere except for a region at the back side of the plate. This means, bidirectional arrows are not suitable if the recognition of regions with compressive and tensile stresses is important. However, bidirectional arrows have the advantage that it is intuitively clear where the stress is compressive or tensile whereas the display using cylinders is of little use without an explanation of the colour coding.

Comparing the displacement direction with the minimum principal stress reveals

---

3Currently this feature is only implemented for planar surfaces in material coordinates.
Figure 5.47. (a) The displacement field (yellow arrows) and the minimum principal stress visualized using bidirectional arrows. Part (b) of the figure shows the same fields but this time the minimum principal stress field is visualized using red and blue coloured cylinders to indicate positive and negative principal stresses, respectively.
that the plate deforms at the back of the hole approximately in the direction of the minimum principal stress (i.e., in the direction of the maximum compressive stress) whereas at the right hand side of the plate it stretches in the direction of the uniaxial load.

**Streamlines**

A continuous representation of a vector field which reduces visual cluttering is given by streamlines. As explained in subsection 4.7.3 a streamline is mathematically described as an integral curve \( x(s) \) which satisfies

\[
\frac{dx}{ds} = v(x(s)) , \quad x(0) = x_0
\]  

(5.4)

where \( v(x) \) is a vector field and the initial condition \( x(0) \) defines the starting point \( x_0 \) of the streamline.

We have implemented three fixed step size ordinary differential equation (ODE) solvers to compute the trajectory \( x \) of a streamline. The most basic ODE solver is the **Euler method** which is derived from the first two terms of the Taylor expansion of \( x(s) \) [eFu]:

\[
x(s_0 + \Delta s) = x(s_0) + \sum_{n=1}^{\infty} \frac{\partial^n x}{\partial s^n} \bigg|_{s_0} (\Delta s)^n n!
\]

Given a start point \( x(s_0) \) the Euler method hence computes the next point on the streamline as

\[
x(s_0 + \Delta s) = x(s_0) + \frac{\partial x}{\partial s} \bigg|_{s_0} \frac{\Delta s}{1} = x(s_0) + \Delta s \, v(x(s_0))
\]

where \( \Delta s \) is called the **time step** of the ODE solver. From the Taylor expansion it can be seen that the error of the Euler method is \( O((\Delta s)^2) \).

Smaller integration errors are obtained with the **mid-point method** which uses the first three terms of the Taylor expansion and has an error of \( O((\Delta s)^3) \) and the **4th order Runge-Kutta method** which uses the first five terms of the Taylor expansion and has an error of \( O((\Delta s)^5) \). We have implemented an OO framework of ODE solvers based on [PVTF92]. It is important to note that the use of higher order solvers with large time steps can lead to large errors around critical points where the field directions and gradients change rapidly. An adaptive time-stepping scheme will be implemented in the future to produce more accurate streamlines in the neighbourhood of critical points.

An example is shown in figure 5.48. The left part of the figure shows the user interface for defining a streamline bundle. The start points of the streamline are specified with the point set selection tool described in subsection 5.6.3. The user must define either a vector field and a scalar field which is colour mapped onto the streamline or the eigenvector field of a tensor field in which case the corresponding eigenvalue is colour mapped onto the streamline. If the vector field is unsigned (e.g.,
Figure 5.48. The user interface for creating a streamline bundle (left) and the results of using different integration methods (right). The yellow streamlines were computed with the Euler method and the red and green streamlines, which are virtually indistinguishable, were computed with the mid-point method and a 4th order Runge-Kutta method, respectively.

an eigenvector field) the streamline is integrated from the start point in both the negative and the positive eigenvector directions.

The bottom section of the user interface is used to specify the integration parameters. Parameters include the ODE solver used, the step size, the maximum number of steps and the maximum length of a streamline. If the latter two parameters are not given the streamline integration continues until the streamline leaves the model domain or until it encounters a zero vector which indicates a critical point in a vector field (see subsection 4.7.8). If a vector field contains circular or vortex patterns this can result in an unacceptably long or even infinite computation.

Examples of the results obtained using the three implemented ODE solvers are shown on the right hand side of figure 5.48. The yellow streamlines were computed with the Euler method and the red and green streamlines were computed with the mid-point method and the 4th order Runge-Kutta method, respectively. In all cases a step size of 0.1 and a maximum of 35 integration steps was used. It can be seen that the streamlines computed with the mid-point method and the 4th order Runge-Kutta method are virtually indistinguishable.

In general we found that the mid-point method is sufficient for smooth trilinearly interpolated vector fields whereas the Euler method produces large integration errors. The Euler method performs poorly because trilinear fields are in general cubic
in any direction not orthogonal to any of the coordinate axes.

Cubic interpolated vector fields and analytically defined vector fields require higher-order ODE solvers such as the 4\textsuperscript{th} order Runge-Kutta method. Note however, that higher order methods with long step sizes can cause problems across element boundaries if the degree of continuity is insufficient.

In addition to the previously explained integration parameters the user can specify a filter (see section 5.7) which terminates the streamline integration if the filter function is not fulfilled. Filters are useful for the visualization of biomedical data sets since they can be used to characterize different tissue types. For example, as explained in subsection 3.3.3 white matter is characterized by a high mean diffusivity and a high diffusion anisotropy. Chapter 7 will demonstrate that nerve fiber tracts can be computed by specifying these conditions as a filter function and by integrating along the major diffusion direction.

**Line Integral Convolution**

In many instances it is desirable to have a dense representation of a vector field over a surface so that all important features are visible. While this can be achieved using streamlines it is difficult to place them appropriately. Subsection 4.7.6 introduced Line Integral Convolution (LIC) as a solution.

We have implemented the original algorithm proposed by Cabral and Leedom.
In order to represent 3D vector fields over 2D surfaces we scale the length of the convolution kernel with the out-of-plane component of the vector field. As a result regions where the vector field is approximately parallel to the surface are represented by long texture components and regions where the vector field is approximately orthogonal to the surface have an almost noise-like texture.

The left part of figure 5.49 shows the user interface for defining a LIC textured surface. The user interface has three sections. The first section is used to select a surface over which the vector field is visualized. The current implementation accepts only material coordinate surfaces. In order to get a smooth polygonal approximation of curved surfaces the user can select the subdivision level.

The second section allows the selection of texture parameters, i.e., the underlying vector field, the size of the texture map (for material coordinate surfaces we have one texture map for each element), and two booleans to enable low pass filtering of the input noise texture and contrast stretching of the output noise texture.

The third and last section of the user interface enables the user to select a scalar field and a colour spectrum in order to colour map the LIC texture.

An example is shown in figure 5.49 on the right. The visualization represents the minor principal stress over the material plane $\xi_1 = 0.5$ of the plate with a hole using line integral convolution. In addition the direction of the minor principal stress is represented by cylinders. It can be clearly seen that the cylinders are aligned with the texture direction and that the texture looks noise like in regions where the direction of the principal stress is almost orthogonal to the surface.

### 5.9.3 Tensor Icons

#### Tensor Ellipsoids

Tensor ellipsoids encode the eigenvectors and eigenvalues of a tensor $T$ by the directions and lengths of its principal axes. As an improvement to the original icon we visualize additionally the sign of each eigenvalue by dividing an ellipsoid into six octagonal sectors which are coloured blue or red if representing a negative or positive eigenvalue, respectively.

The construction of the icon is illustrated in figure 5.50. The unit sphere is divided into 6 octagonal sections orthogonal to the unit axes. Each section is approximated by 16 polygons. The surface normal of each polygon vertex is given by the corresponding point itself. An ellipsoid is obtained by scaling the unit sphere with the diagonal matrix $S$ whose elements are the eigenvalues of the tensor $T$. The ellipsoid is aligned with the tensor’s principal directions by rotating it with the basis transformation matrix $R$ whose columns are the corresponding eigenvectors of $T$. The ellipsoid’s normals are transformed with $(R \cdot S)^{-1}$.

An example is given in figure 5.51. The image illustrates that the maximum principal stress occurs close to the bottom of the back side of the hole and that it is tensile and tangential to the hole’s boundary surface. Therefore the plate is most likely to break under the load at this location. Overall the stress in the plate is in general tensile in the $x$-direction and compressive in the $z$-direction.
Figure 5.50. One octagonal section (red) of a unit sphere (yellow).

Figure 5.51. Tensor ellipsoids visualizing the principal stress in the plate with a hole.
Hyperstreamlines

A continuous representation of a tensor field is obtained by using hyperstreamlines [HD94]. As explained in subsection 4.8.2 the trajectory of a hyperstreamline is a streamline in an eigenvector field. The other two eigenvectors and corresponding eigenvalues define the directions and lengths of the axes of the ellipsoidal cross section of the hyperstreamline.

The user interface for creating hyperstreamline bundles, shown in figure 5.52, is similar to that for creating streamlines. The user must select a tensor field and the eigenvector field which determines the trajectory of the hyperstreamline. The corresponding eigenvalue field is colour mapped onto the hyperstreamline unless the user selects a different scalar field.

Various render options can be used to change the graphical representation of the hyperstreamline. If the “Streamtube” option is selected the diameter of the hyperstreamline is normalised with the maximum transverse eigenvalue. The feature is useful in applications where the transverse eigenvalue fields have large variations. Without normalisation hyperstreamlines can become so narrow that they are invisible or so wide that they obscure other parts of the visualization.

The parameters for streamline integration have been extended by a feature which limits the density of hyperstreamlines. The algorithm subdivides the bounding box of the FE model into a regular grid with a user defined cell size. The user can specify the maximum number of streamlines intersecting a cell or its 6-neighbourhood. If the hyperstreamline intersects a cell whose limit of intersecting hyperstreamlines has been reached the streamline integration is terminated. Using this tool in combination with a regular distribution of sample points creates a dense representation without excessive visual cluttering.

![Figure 5.52. The stress tensor visualized using hyperstreamlines in the direction of the maximum principal stress.](image-url)
5.9 Visualization Icons

Figure 5.52 shows a set of hyperstreamlines in the direction of the maximum principal stress. The hyperstreamlines reveal the complete tensor information along their trajectories with the exception of the sign of the transverse stresses. Perception of the icon shape is improved by illuminating it. However, note that this hampers the correct identification of scalar values from colours. It can be seen that the transverse stresses are oriented approximately in the \( z \)-direction and the \( y \)-direction. The principal stress in the \( z \)-direction is negative (not visible from the visualization) and very small except close to the tip of the hole. The principal stress in the \( y \)-direction changes its sign which is indicated by the constrictions of the hyperstreamlines.

Tensor Topology

Currently we have only implemented a tool for generating the topology of a 2D tensor field. The computation of degenerate points has been described in subsection 4.8.4. The visualization indicates degenerate points by white dots and the direction of the separatrices which separate regions of different behaviour by white arrows. Streamlines are integrated in the direction of each separatrix until they leave the domain or hit another degenerate point.

![Figure 5.53. Topology of a strain field with the maximum principal strain visualized by colour mapped spot noise.](image)

A two-dimensional example of a nonlinear strain field is shown in figure 5.53. In order to enable the reader to compare the tensor topology with the actual tensor field we have visualized the major principal strain using spot noise \[dv95\]. It can be
seen that the separatrices do indeed separate the stress field into regions for which all streamlines in direction of the major principal stress behave similarly. Information about the minor eigenvector field can be obtained by using hyperstreamlines as separatrices.

**Anisotropy Modulated Line Integral Convolution**

In some applications only selected derived measures and not the complete tensor information is required. Subsection 3.3.3 introduced two scalar measures relevant to diffusion tensor imaging data: the mean diffusivity and the diffusion anisotropy, which together with the principal diffusion direction are used to classify neural tissue and to extract nerve fiber tracts.

As a new visualization method for diffusion tensor fields we introduce *Anisotropy Modulated Line Integral Convolution* (AMLIC). The method creates a LIC texture from the maximum principal diffusion direction but additionally defines transparency values inversely proportional to the diffusion anisotropy in order to “remove” regions not corresponding to white matter. The resulting texture is then blended with a colour mapped image of the mean diffusivity using the OpenGL “GL\_BLEND” operation [WND97]. The three dimensional direction of a nerve fiber is encoded by varying the length of a convolution kernel with the normal component of the principal diffusion direction.

The colour coding of the resulting texture provides a segmentation of tissue types and the orientation of the LIC texture indicates the 3D direction of nerve fibers. Chapter 7 gives a more detailed description and an example. The technique can be applied to any tensor field for which the mean eigenvalue and the anisotropy of eigenvalues characterize different regions of the modelled structure.

### 5.10 Additional Features

#### 5.10.1 Contextual Cues

Contextual cues improve perception by enabling the brain to relate abstract visualization icons to familiar objects or properties. An important contextual cue is the model boundary which can be displayed either with shaded surfaces or as a wireframe mesh (see subsection 5.8.1).

In addition the toolkit incorporates labels and markers. These annotations can be used to identify features and to explain relationships.

**Markers**

*Markers* are basic graphic objects with handles for moving and transforming them. Currently implemented markers include spheres to indicate points of interest, 3D arrows to indicate points or forces and other vector information, and B-Spline surfaces, B-Spline curves and polygons to mark areas of special importance. Regions
of interest can be described using spheres and cubes.

![Image of visualization window with markers and labels]

**Figure 5.54.** Example of markers used to highlight model features: The yellow sphere depicts the point of the maximum stress concentration, the blue rectangle the side of the model under uniaxial load and the green arrow indicates the load direction. The image shows additionally a red B-spline surface which is meaningless in this context but can be useful in anatomical models with more irregular features.

Markers are placed and modified using the interaction mechanism described in subsection 5.4.2, i.e., markers have translation handles and modification handles which are rendered as blue and red points, respectively. Translating a translation handle translates the whole object, whereas translating a modification handle only changes that point and so rotates or deforms the object. An example for the application of markers is given in figure 5.54.

**Labels**

Labels are displayed using the `glutBitmapCharacter` function from the GLUT toolkit. A newly created label is placed with its left-top corner at the origin. The user can modify the 3D position of the label using our 3D interaction mechanism. Optionally a label can be displayed in front of the visualization so that it is always readable. In this case the pixel position of the bitmap is determined by projecting the 3D position of the label onto the near plane of the viewing frustum. The contextual information provided by the label is enhanced by an “info” field. An example is shown in figure 5.55.
Figure 5.55. User interface for defining a label (left) and a visualization with two labels shown before (middle) and after (right) rotating the scene.

Anatomical Landmarks

The identification of anatomical structures can be facilitated by inserting easily recognisable features into the 3D visualization. The idea was suggested independently from us by Zhang et al. [ZCML00a] who term these features anatomical landmarks.

Two such structures suitable for visualizing the brain are the ventricles and the eyes (see subsection 7.2.2). The main purpose of inserting anatomical landmarks into a visualization is to give the user an orientation aid when identifying other anatomical features. However, we also found that inserting appropriately chosen landmarks can reduce the amount of visual cluttering as demonstrated in the case studies in the following two chapters. Anatomical landmarks additionally improve 3D depth perception due to interposition (overlay). The effect is increased if shaded landmarks such as isosurfaces are used. If interposition is not desirable semi-transparent surfaces can be employed which we found preferable when using colour mapped surfaces.

5.10.2 Visualization of Raw Data

The understanding and interpretation of a visualization can be improved by incorporating the raw data from which it was derived [SPH+96]. The visualization of raw data was also motivated by discussions with medical specialists who prefer this representation due to its familiarity. Common examples are medical data sets such as MRI, PET, and CT data and electron microscopy images. Our toolkit displays slices of raw data or raw images using texture mapped polygons. Figure 5.56 shows an example.

Additional information can be obtained by processing the raw data (e.g., denoising, contrast enhancement). While computer vision and computer graphics are
5.10 Additional Features

Figure 5.56. The left ventricular FE model of a healthy heart (shown as a wireframe) and one horizontal short axis MRI image from a stack of 8 horizontal images used to construct the FE model.

often considered opposite disciplines [Len98] integrating these two approaches might improve the development and analysis of biomedical models and biomedical visualizations. General issues and applications for a convergence of computer vision and computer graphics are discussed in [LSB00].

5.10.3 Mirrors

Observing a 3D object simultaneously from different sides can not only give additional insight but can also help to better understand its geometry. A natural way to achieve such a display is by placing mirrors into the scene. The human brain is able to assemble the original image and the mirror images of an object into one 3D mental model of it.

Mirror images can be rendered easily in OpenGL by applying an appropriate transformation matrix. A simple example is found in [Kil]. To find the mirror transformation matrix first note that an object is reflected on the \( z = 0 \) plane by the homogeneous reflection matrix \( \mathbf{R} = \text{diag}(1.0, 1.0, -1.0, 1.0) \). Furthermore a point is transformed from the mirror coordinate system into the world coordinate
system using the matrix

\[ M = \begin{pmatrix}
  u_x & v_x & n_x & p_x \\
  u_y & v_y & n_y & p_y \\
  u_z & v_z & n_z & p_z \\
  0 & 0 & 0 & 1
\end{pmatrix} \]

where the orthonormal vectors \( u, v, n \) define the mirror plane and its normal and \( p \) is a point on the mirror plane.

The transformation for reflecting an object is hence given by \( \tilde{R} = MRM^{-1} \).

Note that the light sources have to be reflected as well before rendering the mirror image. A stencil buffer is used to prevent a drawing of the mirror image outside the mirror plane [WND97]. Figure 5.57 demonstrates the usage of mirrors in our toolkit.

Note that we only mirror the model and its visualization but not the mirror planes and mirror images themselves. While this implementation is physically unrealistic it is preferable from a visualization point of view since only information essential for improving the 3D perception is shown and visual cluttering is minimised.

### 5.10.4 Clipping Planes and Sectioning of Models

Structures inside a dataset can be revealed by using clipping planes. Our toolkit provides a simple GUI interface to interactively define clipping planes. The planes are implemented in OpenGL in order to make full use of existing graphics hardware. Consequently the maximum number of simultaneously applied clipping planes depends on the OpenGL implementation used and is limited to six with our current configuration.

Examples are shown in the figures 5.58 and 5.59. The first figure shows the application of multiple clipping planes. Only objects within all of the clipping planes are displayed. The toolkit allows the user to select alternatively a union of clipping planes as shown in figure 5.59. Objects are displayed if they are inside at least one of the clipping planes. The implementation renders the scene multiple times with one clipping plane applied at each iteration. The rendering time using graphics hardware is therefore slightly slower than for the unclipped scene.

An advanced use of clipping planes, which we haven’t seen implemented in any other visualization package, is the simulation of sectioning a data set. The tool might be useful for medical experts who are trained to view multiple sections of a structure and to mentally assemble a 3D view of it. The sectioning process is implemented by using the following algorithm where clipPlane1 and clipPlane2 are initially the cutting planes for the first two sections of the data set.
Figure 5.57. Inserting mirrors into a scene makes it possible to observe different sides of a model simultaneously.
initialise clipPlane1 with specified values;  
clipPlane2 is -clipPlane1 and shifted by sectionDiameter;  
draw model clipped with clipPlane1;  
for(int i=0;i<numSections;i++){  
    if (i>0) shift clipPlane1 and clipPlane2 by sectionDiameter;  
    add translation by shiftVector to modelview transformation;  
    draw model clipped with -clipPlane1 and clipPlane2;  
}  
draw model clipped with -clipPlane2;

Since only two different clipping planes are used in every rendering step an arbitrary number of sections can be defined. If sectioning is enabled the number of clipping planes for conventional clipping is therefore reduced by two. Currently the distance between sections is fixed.

A user defined interaction with sections, such as moving them around or deleting them could be implemented using the tools and techniques described in the previous sections. Also note that medical images are generally viewed on a flat surface with images placed side-by-side. Hence it might be useful to predefine common arrangements of sections rather than letting the user find a suitable arrangement. In our case the potential benefit, however, didn’t warrant the effort.

An example of sectioning a data set is shown in figure 5.60. The isometric arrangement of partially obscured sections shown in the figure is in general not optimal for examining a data set but has has been chosen to demonstrate the sectioning tool.
Also note that clipping planes can still be applied to the sectioned model. Since sectioning the model does not require any geometry changes the clipping operation is performed consistently over all sections of the model.

The tool could be especially useful if employing solid textures (not provided in our OpenGL implementation) and by improving interaction with the sectioning tool, i.e., creating arbitrary cuts and shifting them in space.

### 5.10.5 Input/Output

**Model Description**

Our toolkit handles a variety of models ranging from medical imaging raw data to FE models. In order to handle data sets from different research groups we have provided a front-end to load and translate specific input formats to our internal format. The front-end reads input files that are described in a meta-format indicated by the suffix “.femmodel”. A meta file contains first the type of the model which can be a FE problem (consisting of the FE geometry, material parameters and boundary condition), a FE model (FE geometry and associated fields), a heart model (produced by our collaborators from the Auckland University) or DTI raw data (produced by our collaborators from the NIH). The next line contains the name of the input data set and the following lines contain type specific information.

For example, a FE problem is specified by the name of a file which contains the FE geometry, boundary conditions and material parameters. A left ventricular model is described by a file containing the FE geometry at end-diastole, a file with bezier surface representations of the endocardial and epicardial surface (which are used to compute the element scale factors in circumferential and longitudinal direction), a file with the myocardial strain field and a file with the FE geometry at end-systole. Figure 5.61 shows several examples of model meta-files.
Figure 5.61. Several examples of models described using a meta-format.
Visualization I/O

Creating a good visualization can be challenging and time consuming. It is therefore desirable to be able to save and load a visualization so that it can be easily reproduced if needed.

A visualization icon can be saved either by specifying the type of the icon and a list of parameters necessary to recompute it or by additionally saving the list of graphic primitives representing the icon.

An isosurface, for example, can be described by the field name, the isovalue, the number of subdivisions, the surface colour or the colour map used, and boolean values specifying the five rendering options listed in its GUI. The full graphic representation would save additionally the list of polygons representing the isosurface.

The first choice has the advantage of minimal storage requirements and a greater speed if the recomputation of the graphic primitives is faster than the reading time from the storage medium. The second alternative provides immediate display of the graphic once it is loaded. This alternative is recommendable for visualization icons which require a long time to compute but have a manageable amount of graphic primitives. Currently these option are only partially implemented. Similarly we haven’t implemented yet a tool to save and load derived fields and volume, surface and point selections.

A possible alternative or addition to the currently implemented technique is scripting. With scripting every action in a session is recorded to a script file that is saved at the end of the session. Each and every action, be it a change in the lightning model or the position or type of icons, is recorded as a simple text string to keep the file size down. By replacing the script file at the start of a new session, it is possible to re-construct the state of a visualization as it stood at the end of the previous session. Furthermore a scripting language enables the user to go backward and to make modifications to a visualization or to apply an existing script to a new data set [Ken04].

VRML Output

An important feature of any visualization package is the communication of the visualization results with outside users such as other research groups and medical personal. We achieve this by providing output in VRML-format. VRML stands for Virtual Reality Modelling Language and is an internet standard for 3D graphics. VRML files can be viewed over the Internet using Web3D viewers which usually come as a plug-in to Internet browsers. Figure 5.62 shows a brain dataset viewed with an Internet browser and the Web3D viewer Cortona which is available for free over the internet [Par02].

A future extension is to store a visualization in XML format. The bioengineering group of the University of Auckland recently devised such a format for finite element models.

Note that the VRML format also is ideally suited when employing visualizations as a teaching tool.
Figure 5.62. A screen shot of the visualization of the diffusion tensor field in the brain viewed with a Web3D viewer over the internet.
Output for Direct Volume Rendering

A popular method for displaying 3D scalar fields is direct volume rendering. We found that direct volume rendering offers little additional benefit when visualizing tensor fields. However, if desired our toolkit allows the output of scalar fields in a format suitable for a public domain interactive volume visualization program developed by Rezk-Salama et al. from the University of Erlangen-Nürnberg and the University of Stuttgart [RSEB+00]. The volume visualization software makes use of the multi-texturing capabilities of the GeForce graphics card and allows interactive frame rates on a standard PC.

A volume visualization input file is created by sampling a scalar field at regular sampling points over the bounding box of the model or a suitable user defined volume. An example of the resulting visualizations is shown in figure 5.63.

Figure 5.63. A screen shot of the visualization of the mean diffusivity in the brain using a public domain volume visualization software.
5.11 Rendering Control

The rendering control contains the view parameters, a trackball, lighting information, mirrors, and global clipping planes. The same rendering control can be used for different windows which is practical, for example, when comparing two different models.

A useful feature is the animation of models. Animations are valuable when using large numbers of icons distributed over a 3D domain. Rotating the model around its axis enables the user to differentiate icons in the foreground and background. The toolkit contains a feature to animate the trackball used to rotate the model. A fly-through is also available.

It is important to mention that all visualization icons have a default setting specifying whether they are illuminated or not. The user can change this setting if required. For most visualization icons lighting is enabled since shading is an important shape cue in 3D vision [WL01a]. However, illuminating a surface makes it difficult to perceive the object colour so that lighting is disabled for colour mapped objects as long as they are flat or sufficiently smooth. Light positions, light amplitude and diffuse, ambient and specular light components can be changed interactively and multiple lights can be defined.

5.12 Conclusion

This chapter described a visualization toolkit for biomedical data sets with several novel features: The first feature is a modular design with separate objects describing the underlying model, the visualization including data fields, rendering parameters, and visualization windows. A visualization is achieved by defining relationships, subject to some constraints, between these objects. The design facilitates the definition of simultaneous visualizations of multiple models such as the simultaneous display of a sick and a healthy heart or the simultaneous display of a global and a detailed view of an object. Using the same rendering parameters ensures that both models are displayed using the same view, scaling, orientation and lighting.

The second novel feature is a generalized field data structure. The user can define scalar fields, vector fields, and tensor fields over a FE mesh using suitable interpolation functions, by defining regular samples and a reconstruction filter or as analytic functions. The fields can be used to derive new fields using a set of predefined operators and general algebraic expressions. Examples are gradient fields and eigenvalue and eigenvector fields. The advantage of this construction is that fields are only evaluated if they are used and that the user can interactively construct new non-standard fields when required for a given application. We found the feature especially useful when exploring tensor fields since the formation of new measures such as the diffusion anisotropy can be used to extract anatomical structures. Fields can be represented with respect to both material and world coordinates.

The third novel feature of our toolkit are boolean filters which are used to control the positioning and shape of visualization icons. Boolean filters are also used for the
creation of conditional fields which can be used for the segmentation of data.

Lastly the toolkit features a global colour map control and a model dependent point, surface and volume selection mechanism. The global colour map was motivated by the observation that users find it easier to derive qualitative and quantitative information when using the same colour scale for different fields in different models. Defining new colour maps is often necessary to avoid colour clashes when displaying multiple visualization icons simultaneously and gives the user additional freedom when exploring a data set. As a novel modification we suggested spectrum markers, a technique to add isocontours to a colour map, and cyclical colour maps which are useful to extract structure and symmetries from fields.

Model dependent point, surface, and volume selection mechanism facilitate the placement and mixing of visualization icons. We also suggested some minor improvements to existing icons and we introduced Anisotropy Modulated Line Integral Convolution as a new visualization method for tensor fields.

We used our visualization toolkit successfully to explore tensor fields in the heart and the brain [WL01b, Wün03b, WY03] as described in the following two chapters.

5.13 Future Work

Much work remains to be done in order to obtain a professional easy-to-use visualization environment. One feature we want to implement in the future is to make isosurfaces a part of the surface selection tool. Isosurfaces could then be colour mapped or used for creating surface-based sample point sets. Note that sample points can be defined regularly over a general non-parameterized surface by using a point repulsion mechanism [Tur91, Tur92]. More advanced methods for resampling point sampled surfaces are described and analysed in [PGK02].

A general difficulty with streamlines is the choice of appropriate points to start and terminate the streamlines. Streamlines started at equidistant locations can quickly diverge or converge resulting in visual cluttering or large uncovered areas. A simple solution is to split a streamline if the divergence exceeds a certain predefined limit and to terminate it if the convergence surpasses a given value. In order to prevent all streamlines being split or terminated at once a probabilistic decision can be taken. Several algorithms for streamline placement are described in the literature [TB96, MHHI98].

Currently we have implemented only algorithms for computing the 2D tensor topology. In order to further the exploration of biomedical tensor fields we intend to implement in future algorithms for computing the 3D tensor topology.
This case study demonstrates how our toolkit can be used to measure and visualize left ventricular deformation in order to obtain an improved understanding of cardiac mechanics. We use as examples a left-ventricular finite element model (see subsection 3.2.8) of a healthy heart and a FE model of a heart diagnosed with non-ischemic dilated cardiomyopathy, a non-ischemic disease or defect of the heart muscle which impairs its pumping action. The disease is characterized by an enlargement of the weakened heart muscle leading to inefficient contraction of the pumping chamber and to heart failure [Maya].

The following contributions are made: we apply techniques traditionally used in solid mechanics and computational fluid dynamics to biomedical data and suggest some improvements and modifications. We obtain new insight into the mechanics of the healthy and the diseased left ventricle and we facilitate the understanding of the complex deformation of the heart muscle by novel visualizations.

In the following the strain tensor $\mathbf{E}$ is defined with respect to the material coordinates of the left-ventricular finite element model introduced in subsection 3.2.8. The normal components $E_{11}$, $E_{22}$, and $E_{33}$ represent therefore the strains in the circumferential, longitudinal, and radial directions, respectively. This approach gives better results than obtained by using the normal components of a cylindrical strain tensor as done in [GZM97][You00].

For comparison we visualize the left ventricle of the diseased heart together with that of a healthy heart. Unless noted otherwise the images show the healthy heart on the left hand side and the diseased heart on the right hand side.
6.1 Computing Ventricular Performance Measures

The performance of the left ventricle is often specified using various length, surface and volume measures such as its systolic and diastolic volume and its ejection fraction. Using our visualization toolkit the user can specify elements, faces and parameter curves and compute their volume, area and length, respectively [Wun03b, WY03].

6.1.1 Computing Volume Measures

The volume of the heart muscle can be computed using the technique explained in subsection 5.8.2. Table 6.1 shows the myocardial volume at end-diastole and end-systole and the resulting volume reduction during contraction. In general the myocardium is considered incompressible but Denney and Prince estimate that small volume changes up to 10% occur due to myocardial perfusion [DP95]. Our results show considerably higher values for the healthy heart. A possible explanation is that the wall thickening strain appears to underestimate the actual strain. We believe this is due to the fact that thickening increases dramatically toward the endocardium (due to the nearly incompressible nature of the muscle) and the tag resolution of two or three stripes across the wall (see figure 3.10) is inadequate to capture this [You02]. As a result the computed displacements underestimate the actual tag line displacements in the endocardium and hence the computed wall thickening and the myocardial volume at end-systole are underestimated.

<table>
<thead>
<tr>
<th></th>
<th>ED</th>
<th>ES</th>
<th>Myocardial volume reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Healthy heart</td>
<td>217.6</td>
<td>159.1</td>
<td>26.88%</td>
</tr>
<tr>
<td>Sick heart</td>
<td>336.7</td>
<td>305.3</td>
<td>9.32%</td>
</tr>
</tbody>
</table>

Table 6.1. Myocardial volume (in cm$^3$) of the healthy and the diseased left ventricle at end-diastole (ED) and end-systole (ES).

One of the most important measures of cardiac performance is the ventricular (blood) volume and the fraction of blood ejected during contraction. In order to apply the volume computation introduced in subsection 5.8.2 the left-ventricular cavity must be modeled by finite elements. Using our toolkit we can define centroids for any four vertices on the endocardial surface with common longitudinal $\xi$-coordinate. Connecting these vertices to the corresponding points on the endocardial surface results in 16 finite elements for the left ventricular cavity.

Figure 6.1 and 6.2 show the finite element models of the left ventricular cavity of the healthy and the sick heart, respectively, at end-diastole and end-systole.

Using equation 5.1 we can now compute the left ventricular volumes at end-diastole (ED) and end-systole (ES). The difference of these values represents the
6.1 Computing Ventricular Performance Measures

Figure 6.1. Left ventricular cavity of the healthy heart at end-diastole (left) and end-systole (right).

Figure 6.2. Left ventricular cavity of the sick heart at end-diastole (left) and end-systole (right).
stroke volume [volume of ejected blood] (SV) and the ratio of stroke volume to the volume at end-diastole represents the ejection fraction (EF). The results for the healthy and the diseased heart are shown in Table 6.2.

<table>
<thead>
<tr>
<th></th>
<th>Healthy heart</th>
<th>Sick heart</th>
</tr>
</thead>
<tbody>
<tr>
<td>ED</td>
<td>87.15</td>
<td>314.18</td>
</tr>
<tr>
<td>ES</td>
<td>35.08</td>
<td>277.94</td>
</tr>
<tr>
<td>SV</td>
<td>52.07</td>
<td>36.23</td>
</tr>
<tr>
<td>EF</td>
<td>59.75%</td>
<td>11.53%</td>
</tr>
</tbody>
</table>

Table 6.2. Ventricular volume (in cm$^3$) of the healthy and the diseased left ventricle at end-diastole (ED) and end-systole (ES), stroke volume (SV), and ejection fraction (EF).

The ventricular volume of the healthy heart at end-diastole is about 87cm$^3$ and the stroke volume is 52cm$^3$ resulting in an ejection fraction of about 60%. These values correspond well with data reported in the medical literature [Box99, LSM+02]. We think that the values slightly underestimate the actual ejection fraction due to the difficulties with computing the radial strain. The current model does not track tags at the endocardial boundary which might give a better approximation of inner wall motion.

For the diseased heart a considerably larger end-diastolic volume is observed. However, the stroke volume is only 36.23cm$^3$ and about 30% smaller than for the healthy heart. The ejection fraction is only 11.5%. These values indicate a severe impairment of myocardial function.

### 6.1.2 Computing Ventricular Surface Areas

Table 6.3 shows the areas of the endocardial and the epicardial surface computed using equation 5.2. It can be seen that the area reduction of the sick left ventricle is severely impaired. Since the muscle fibers of the myocardium are aligned with these surfaces the measurements indicate that either muscle fibers don’t contract (e.g., due to fibrosis) or that they contract in some regions but expand in other regions of the surface. In order to further examine this deformation behaviour we will visualize the strain tensor in the next section.

<table>
<thead>
<tr>
<th></th>
<th>ED</th>
<th>ES</th>
<th>Area reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epicardial surface healthy heart</td>
<td>201.7</td>
<td>147.7</td>
<td>26.75 %</td>
</tr>
<tr>
<td>Endocardial surface healthy heart</td>
<td>93.4</td>
<td>53.4</td>
<td>42.76 %</td>
</tr>
<tr>
<td>Epicardial surface sick heart</td>
<td>350.6</td>
<td>324.6</td>
<td>7.40 %</td>
</tr>
<tr>
<td>Endocardial surface sick heart</td>
<td>218.7</td>
<td>200.0</td>
<td>8.55 %</td>
</tr>
</tbody>
</table>

Table 6.3. Surface area (in cm$^2$) of the endocardial and the epicardial surface of the healthy and the diseased left ventricle at end-diastole (ED) and end-systole (ES).

Using the technique introduced in subsection 5.8.2 it is also possible to compute the midventricular cavity cross-sectional area. We get as results 13.27cm$^2$ at
end-diastole and 5.81cm² at end-systole. From these values we determine a mid ventricular radius of 2.06cm at end-diastole and 1.36cm at end-systole which lies within the normal range of values [You02].

### 6.1.3 Computing Length Measures

We compute the circumference of a short-axis cross section of the left ventricular cavity by the length of a curve on the endocardial surface with a constant longitudinal \( \xi \)-parameter. The length of this curve can then be used to derive a value for the ventricular radius at that position. However, reliable results are only obtained if the arc is approximately planar and orthogonal to the long axis of the ventricle. While the technique could also be used to approximate the wall thickness at a point (by computing the arc length in radial direction) this does not necessarily yield the shortest distinct between the endocardial and epicardial surface. Better computational techniques are suggested in [vR99] and alternative results are given in [KPF*99].

### 6.2 The Visualization of Myocardial Strain

The measurements presented in section 6.1 indicate a severe impairment of the contraction of the sick heart. In order to better understand the local deformation of the myocardium more information is required. This section presents and explains various visualizations of the strain tensor and of quantities derived from it [WY03, WLY04]. Most visualization methods in this section visualize the strain tensor by using its principal directions and principal strains explained in subsection 2.3.1.

#### 6.2.1 Tensor Ellipsoids

As an initial visualization we display tensor ellipsoids at regular sample points throughout the midmyocardium. As explained in subsection 5.9.3 tensor ellipsoids encode the principal directions and strains by the directions and lengths, respectively, of the axes of the ellipsoid. The segments of the ellipsoids are coloured according to the sign of the principal strains with red indicating expansion and blue indicating contraction. Note that the 3D geometry is difficult to perceive from a static image. Rotating the model enables the brain to differentiate ellipsoids in the foreground and background.

Figure 6.3 shows that for the healthy ventricle the myocardium expands in the radial direction (wall thickening) and contracts in the longitudinal and circumferential direction with the circumferential contraction being in general larger. The contraction is smallest in the septum and largest in the free wall. The results correspond well with measurements reported in the literature [LOMP+94, GZM97, YICA94, DM97].

The deformation of the sick ventricle is highly abnormal. Whereas the anterior-lateral wall of the ventricle displays an almost normal deformation behaviour, albeit
with smaller strain values, the situation is the exact opposite in the septal wall of the ventricle. Here the myocardium is contracting in the radial direction and is expanding in the circumferential and longitudinal direction.

### 6.2.2 Streamlines

While each tensor ellipsoid displays the complete tensor information at a point the resulting visualization suffers from visual cluttering. A continuous representation of a vector field (e.g., an eigenvector field) along a line is obtained by using streamlines which are at each point tangential to the underlying vector field.

Figure 6.4 uses colour mapped streamlines to visualize the direction and magnitude of the major principal strain. Note that an eigenvector field is unsigned (i.e., eigenvectors have a direction but not an orientation) and that therefore the streamlines must be integrated in both the positive and the negative direction of the eigenvector field.

Streamlines are rendered as thin tubes with a constant diameter rather than as lines. Illuminating these tube-like structures gives important shape and depth cues which aid their 3D perception [WL01a]. We also render the endocardial wall (in gray) in order to reduce visual cluttering caused by the overlap of streamlines in the foreground and the background.

The image on the left of figure 6.4 shows clearly that for the healthy heart the major principal strain is oriented in the radial direction throughout the myocardial
Figure 6.4. The myocardial strain field in the healthy (left) and the diseased (right) left ventricle visualized using streamlines in the direction of the major principal strain. The septal wall is indicated by a yellow sphere.

wall and that it is positive and increases toward the endocardium. This observation is consistent with an increased wall thickening towards the endocardium.

The image on the right of figure 6.4 confirms the previously identified abnormal contraction of the diseased left ventricle. The direction of the major principal strain is normal in the anterior-lateral and the inferior-lateral wall. However, the magnitude of the major principal strain in the inferior-lateral wall is considerably smaller than for the healthy heart and is negative in some regions (indicating a wall thinning instead of a wall thickening). In the septal wall of the diseased heart the maximum principal strain is oriented in the longitudinal and circumferential directions rather than in the radial direction.

6.2.3 Hyperstreamlines

Streamlines encode only one eigenvector. A continuous representation of the complete strain tensor along a line is achieved by using hyperstreamlines [DH93].

The trajectory of a hyperstreamline is a streamline in an eigenvector field. The other two eigenvectors and corresponding eigenvalues of the strain tensor define the axes and lengths of the ellipsoidal cross section of the hyperstreamline. The remaining eigenvalue is colour mapped onto the hyperstreamline.

Figure 6.5 and 6.6 show hyperstreamlines in the direction of the major and minor principal strain, respectively. The image on the left of figure 6.5 shows again that for the healthy heart the major principal strain is oriented in the radial direction.
Figure 6.5. The strain field in the healthy (left) and the diseased (right) left ventricle visualized using hyperstreamlines in the direction of the major principal strain. The septal wall is indicated by a yellow sphere.

Figure 6.6. The strain field in the healthy (left) and the diseased (right) left ventricle visualized using hyperstreamlines in the direction of the minor principal strain. The septal wall is indicated by a yellow sphere.
6.2 The Visualization of Myocardial Strain

throughout the myocardial wall and that it is positive and increases toward the endocardium. Furthermore it can be seen from the diameter of the cross section of the hyperstreamline that with the exception of the septal wall the magnitude of the transverse strains increases from the epicardial to the endocardial surface. We are not aware of any previous work showing all these properties with a single image.

Figure 6.7. The strain field in the healthy (left) and the diseased (right) left ventricle visualized using hyperstreamlines in the direction of the minor principal strain. Perception of the complex 3D geometry is improved by rendering the endocardial wall in gray and by inserting mirrors into the scene. The septal wall is indicated by a yellow sphere.

The perception of the hyperstreamlines in the direction of the minimum principal strain, shown in figure 6.6, is straightforward when using an animated visualization but it is difficult when using the given figure alone. Figure 6.7 shows an improved version of this visualization. Perception is enhanced by rendering the endocardial surface in gray. The occluded portion of the visualization is revealed by using two mirrors.

The figure shows that the minimum principal strain of the healthy left ventricle is compressive throughout most of the myocardium and its direction over most of the myocardium resembles a spiral moving toward the apex. This strain direction corresponds well with the motion of the heart described in the medical literature: the septum initially performs an anticlockwise rotation (apex-base view) but later a more radial movement. The apex rotates overall anticlockwise whereas the base rotates clockwise. The anterioseptal regions of the mid and apical levels and the posteroseptal region of the base perform a hook-like motion because of a reversal of rotation [YICA94]. Note that we have in the posteroseptal region and the anterioseptal region one interesting feature where the hyperstreamlines change suddenly
The image on the right hand side of figure 6.7 illustrates again that the minor principal strain in the lateral wall of the diseased heart (right mirror image) is similar to that of the healthy heart. In contrast to this the minimum principal strain in the septal, posterioseptal (left mirror image) and the anterioseptal wall (facing the viewer) points in the radial direction. Note that this is highly abnormal since for the healthy heart the radial direction is aligned with the direction of the maximum principal strain (see figure 6.5).

6.2.4 Line Integral Convolution

The above described features where hyperstreamlines change direction can be examined in more detail using a line integral convolution texture. We use the direction of the minor principal strain as a vector field and use its magnitude to colour map the texture.

Figure 6.8 shows that the maximum compressive strain in the midmyocardium is predominantly oriented in the circumferential direction with a slight downward tilt. Several interesting points exist where the strain suddenly changes direction. Results from tensor analysis show that these points are degenerate points for which at least two eigenvalues are equal [DH94]. Two such points are indicated by magenta coloured disks in the enlarged region shown on the right hand side of the top image. We found that most of the degenerate points occur on or near the septal wall. The unusual variations in strain orientation might be caused by the right ventricular wall which is connected to the left ventricular wall at both sides of the septum.

In contrast to the healthy heart the strain field of the sick heart contains considerable more degenerate points distributed throughout the myocardium. The enlargement on the right hand side of the image in the middle shows the presence of a line for which each point on it is a degenerate point. The differences are especially striking when comparing the lateral wall of the healthy and the sick left ventricle shown in the two images at the bottom of figure 6.8.

6.2.5 Colour Mapped Surfaces and Isosurfaces

We conclude this section with an examination of the distribution of the strains in the material directions. Since the strain tensor is defined with respect to the material coordinates the strains in the circumferential, longitudinal and radial directions are given by the normal components \( E_{11} \), \( E_{22} \) and \( E_{33} \), respectively, of the strain tensor \( E \).

Figure 6.9 visualizes the normal strains on the endocardial surface using colour mapping and shows additionally the 0-isosurface, which separates contracting and expanding regions. The isosurfaces were computed with the modified Marching Cubes algorithm described on page 172 of subsection 5.9.1.

The images on the left of the figure show clearly that the healthy left ventricle contracts in the circumferential and longitudinal directions and expands in the radial
Figure 6.8. The minor principal strain (maximum contracting strain) in the healthy (top) and the sick (middle) heart visualized using Line Integral Convolution. The bottom images show the lateral wall of the healthy (left) and the sick (right) heart. The magenta coloured disks and lines indicate degenerate points.
Figure 6.9. The normal strain in the circumferential (top), longitudinal (middle) and radial (bottom) direction on the endocardial surface of the healthy (left) and the sick (right) heart visualized using colour mapping. The images show also the 0-isosurface which separates regions of contractile and expanding strain. The septal wall is indicated by a yellow sphere.
direction. The only exceptions are some parts of the model boundary at the base and, for the radial strain, three small cylindrical regions at the apex and the septal and lateral wall. All three normal strain components are distributed relatively evenly over the endocardial surface.

For the diseased heart the lateral wall and part of the anterior and inferior wall contract in the circumferential and longitudinal directions. Wall thickening is observed in the basal-lateral wall, the basal-septal wall and in parts of the anterior and inferior wall. The rest of the myocardium shows an abnormal deformation. As a result of the strain distribution the ventricle does not contract evenly but rather performs a shape change.

We are also interested in the shear components of the strain tensor. It is known that during contraction the heart changes predominantly in diameter. LeGrice et al. [LTC95] report 8% lateral expansion but 40% wall thickening. This indicates reorganization of the myocytes during systole. Because of the sheet structure of the myocardium it has been proposed that the sheets can slide over one another restricted mainly by the length of the interconnecting collagen fibers [LTC95]. The shear properties of the myocardium resulting from this sliding motion are characterized in [DLS+00, DSYL02]. The shear is most restricted in the direction of the sheet normals and the maximum shear is possible in the fiber direction. Wall shear is thought to be an important mechanism of wall thickening during systole and therefore may play a substantial role in the ejection of blood from the ventricle.

Figure 6.10 shows a visualization of the circumferential-longitudinal shear strain (component $E_{12}$ of the strain tensor) on the endocardial surface of the healthy (left) and the sick (right) heart visualized using colour mapping. For the healthy heart the shear strain is positive for most of the myocardium with the exception of some subepicardial regions close to the merging point with the right ventricular wall. No consistent behaviour can be found for the diseased heart. The shear in the lateral wall resembles most closely the normal range of values whereas the anterior-basal region exhibits extremely high negative strains, which might indicate impending tissue damage.

6.3 Displacement Field Visualization

The movement of the heart during contraction can be studied by visualizing the displacement field between end-diastole and end-systole. Figure 6.11 indicates the displacement at selected material points with red arrows. It can be seen that the heart moves during contraction towards the apex.

In order to analyse rotational movements we project the displacement vectors onto a radial-circumferential material plane. The results are visualized in figure 6.12 and indicate that the apex rotates overall anticlockwise (apex-to-base view) whereas the base moves in radial direction.
**Figure 6.10.** The circumferential-longitudinal shear strain on the endocardial surface of the healthy (left) and the sick (right) heart visualized using colour mapping. The images show additionally the 0-isosurface of this strain component. The septal wall is indicated by a yellow sphere.

**Figure 6.11.** The displacement field of the contracting left ventricle visualized using vector arrows.

**Figure 6.12.** The displacement field of the contracting left ventricle visualized using vector arrows projected onto a radial-circumferential material plane.
6.4 Conclusion

Visualizing the strain field improves the understanding of the complex deformation of the heart muscle. Using techniques new to the biomedical field offers additional insight. The visual information can be supplemented by computing ventricular performance measures which are easily obtained from the finite element model using numerical integration.

The visualization of the healthy heart confirms observations previously reported in the literature. Using tensor ellipsoids, streamlines and hyperstreamlines makes it possible to visualize complex deformation behaviour in a single image. Line integral convolution uncovers the presence of degenerate points at which the principal strains suddenly change direction. Further investigations are necessary to find the relationship between degenerate points, fiber structure, and the ventricular anatomy. Furthermore we want to explore their significance (if any) for diagnosing heart diseases.

Visualizing a ventricle with dilated cardiomyopathy showed that the deformation of the lateral wall is roughly like that of a normal heart whereas the septal wall behaved almost in the opposite manner. Very large negative shear strains were recorded in the anterior-basal wall of the ventricle. The combined effect of these deformations seems to be a pumping action by shape deformation (from a circular to an ellipsoidal cross section) rather than by contraction.

The visualizations and measurements performed in this case study demonstrate the usefulness of our visualization toolkit for exploring biomedical models. Using the unique field data structure enables the interactive definition of new measures and facilitates the exploration of the data set. The modular OO-design allows comparison of multiple models, which is further enhanced by the user interface for colour map design and control.

6.5 Future Research

We are interested in visualizing other data sets of diseased hearts, in particularly models of ischemic myocardium. It is known that small changes in the deformation behaviour of the myocardium occur before the first symptoms of a cardiac infarct develop and we hope that visualizing myocardial strain supports the detection of regions of low blood perfusion. Non-traditional visualization methods such as hyperstreamlines, LIC and tensor topology seem to be particularly promising for this purpose.

Of particular interest is the relationship between myocardial strain and fiber structure. Recent research suggests that the measurement of the fiber structure is possible using diffusion tensor imaging [SHS+01, MFE+01, ACCM01]. Further information could be provided by fusing our data with functional data obtained by PET and SPECT [RdB99].

Of additional interest is the comparison of the myocardial deformation with the blood flow pattern in the coronary vessels. Earlier work by Kilner et al. [KYW+00]
visualized the blood flow on each MRI plane using streamlines. The authors suggest that the asymmetries and curvature of the heart have potential fluid dynamic advantages and show an improved ventriculo-atrial coupling if compared with a simpler atrio-ventricular arrangement as found in snails. It has been shown that atherosclerotic plaques develop in regions where flow rate and shear strain are relatively low and that regions of low shear strain are pathologically prone to vessel wall thickening and thrombosis [Bro00b].

We hope that in future visualizing different cardiac data sets simultaneously will further improve the understanding of the heart and the diagnosis and treatment of cardiovascular diseases.
A common problem in biomedical sciences is the in vivo measurement of anatomical structures. This case study demonstrates how tissue types, nerve fiber tracts, and differences in the structure of nerve fiber tracts in the brain are visualized from diffusion-weighted MRI data using our toolkit.

The diffusion tensor describes the spatial distribution of water molecules originating at a common location. Since the diffusion of water depends on the microstructure of the tissue, the diffusion tensor field can be used to visualize nerve fibers and other tissues in the brain. The resulting images of the brain anatomy can be used to advance research in surgical planning, cognitive sciences [PMF98], and the diagnosis and treatment of various white and gray matter disorders [PJB96, Hedb, Heda, BGEH03]. Visualizing the nerve fiber structure also represents a valuable teaching tool.

The anatomy of the brain and diffusion tensor imaging (DTI) have been introduced in section 3.3. This chapter starts with an introduction of several new diffusion measures and then describes an incremental approach for the visual exploration of diffusion tensor data. Our approach starts with slice images familiar to the medical specialist and progressively expands the dimension and abstraction level of the representation in order to provide new insights into the data. In particular we present three new techniques for the visualization of DTI data. Barycentric colour maps allow an integrated view of different types of diffusion anisotropy. Ellipsoid-based textures in combination with barycentric or spherical colour maps indicate the nerve fiber direction and different anisotropy properties or tissue types, respectively. Anisotropy Modulated Line Integral Convolution (AMLIC) creates an image segmented by tissue type and incorporating a texture representing the 3D orientation of nerve fibers. The quality of our exploration approach and new visualization techniques are demonstrated by identifying various anatomical structures and features in a diffusion tensor data set of a healthy brain.
7.1 Diffusion Measures

To facilitate the definition of diffusion measures we order the three eigenvalues of the diffusion tensor \( \mathbf{D} \) by size with \( \lambda_1 \) being the biggest and \( \lambda_3 \) being the smallest.

The two most important measures are the mean diffusivity and the diffusion anisotropy. As explained in subsection 3.3.3 the mean diffusivity is defined as

\[
\lambda_{\text{mean}} = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3} = \frac{\text{trace}(\mathbf{D})}{3} = \frac{D_{11} + D_{22} + D_{33}}{3}
\]

We use as a measure for the diffusion anisotropy

\[
\lambda_{\text{anisotropy}} = \frac{\text{trace}((\mathbf{D} - \lambda_{\text{mean}} \mathbf{I})^2)/\lambda^2_{\text{mean}}}{(\mathbf{D} - \lambda_{\text{mean}} \mathbf{I})^T(\mathbf{D} - \lambda_{\text{mean}} \mathbf{I})}/\lambda^2_{\text{mean}}
\]

which was suggested to us by Peter Basser [Bas00]. Note that the computation is efficient and does not require the computation of the eigenvalues. The measure is used to identify regions where \( \lambda_1 \gg \lambda_2 \geq \lambda_3 \) and can be used to detect nerve fiber tracts.

We are also interested in regions with a planar diffusion behaviour, i.e., regions with \( \lambda_1 \geq \lambda_2 \gg \lambda_3 \) (oblate diffusion). A special case are regions where additionally \( \lambda_1 \approx \lambda_2 \) (circular oblate diffusion). Regions of oblate and circular oblate diffusion indicate a crossing of nerve fibers tracts or white matter fibers randomly oriented on a plane [PJB+96].

In order to define new measures note that the ratio of any two diffusions in orthogonal directions becomes maximal and minimal for the principal diffusion directions. Since \( \lambda_2 \) and \( \lambda_3 \) are the largest and smallest diffusivities, respectively, orthogonal to the direction of the maximum principal diffusion we define

\[
\lambda_{\text{anisotropyRatioMax}} = 1 - \frac{\lambda_3}{\lambda_1} = \frac{\lambda_1 - \lambda_3}{\lambda_1}
\]

\[
\lambda_{\text{anisotropyRatioMin}} = 1 - \frac{\lambda_2}{\lambda_1} = \frac{\lambda_1 - \lambda_2}{\lambda_1}
\]

\[
\lambda_{\text{traverseAnisotropy}} = 1 - \frac{\lambda_3}{\lambda_2} = \frac{\lambda_2 - \lambda_3}{\lambda_2}
\]

Above measures have values in the range \([0, 1]\) and are one if the anisotropy is maximal. Inverse measures can be analogously constructed, e.g.,

\[
\hat{\lambda}_{\text{traverseAnisotropy}} = 1 - \lambda_{\text{traverseAnisotropy}} = \frac{\lambda_3}{\lambda_2}
\]

By taking the product of several of these measures it is now possible to create expressions for the oblate anisotropy and the circular oblate anisotropy. The oblate anisotropy is characterised by a high anisotropy ratio with respect to the minimum diffusion direction, i.e.,

\[
\lambda_{\text{oblateAnisotropy}} = \frac{\lambda_1 - \lambda_3}{\lambda_1} \frac{\lambda_2 - \lambda_3}{\lambda_2} = \frac{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)}{\lambda_1 \lambda_2}
\]
The circular oblate anisotropy requires additionally that the maximum and minimum diffusivities are similar, i.e.,

\[
\lambda_{\text{circularOblateAnisotropy}} = \frac{\lambda_2}{\lambda_1} \lambda_{\text{oblateAnisotropy}} = \frac{\lambda_2 \lambda_1 - \lambda_3 \lambda_2 - \lambda_3}{\lambda_1} = \frac{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)}{\lambda_1^2}
\]

Both measures are normalised, i.e., have values in the range [0, 1], are equal if \(\lambda_1 = \lambda_2\) and have the property that

\[
\lambda_{\text{circularOblateAnisotropy/oblateAnisotropy}} = \begin{cases} 
0 & \text{if } \lambda_1 = \lambda_2 = \lambda_3 \\
1 & \text{if } \lambda_1 = \lambda_2 \neq \lambda_3 = 0 \\
\left(\frac{n-1}{n}\right)^2 & \text{if } \lambda_1 = \lambda_2 = n\lambda_3 \neq 0
\end{cases}
\]

### 7.2 Exploration of a Diffusion Tensor Data Set

This section presents a step-by-step exploration of diffusion tensor data starting with simple slice images and culminating with complex 3D representations of nerve fiber tracts.

#### 7.2.1 Colour Mapped Surfaces

Traditionally MRI data is displayed as a set of slices parallel to one of the coordinate planes. The optimal presentation of MRI slice images for improved perception of information has been researched by [vCIA98, vAIC99].

As a basic example for colour mapping consider the figures 7.1–7.3 which show nine equidistant horizontal slices through the brain from top to bottom colour mapped with the maximum diffusivity \(\lambda_1\), the mean diffusivity \(\lambda_{\text{mean}}\), and the diffusion anisotropy \(\lambda_{\text{anisotropy}}\), respectively. The images are ordered from left to right and top to bottom.

It can be seen that naive use of colour mapping does not differentiate anatomical structures from the background which makes the interpretation and analysis of the visualization difficult. For the remainder of this chapter we classify background voxels by identifying an appropriate cut-off value from the histogram of the mean diffusivity of the image data set.

The horizontal maps of the maximum diffusivity (figure 7.1) indicate fluid filled compartments in saturated blue, red or yellow and fiber tracts in saturated blue which makes it virtually impossible to differentiate it from cerebral spinal fluid (CSF). Gray matter has a very low maximum diffusivity and is indicated in blue-black.

An improved contrast between CSF and fiber tracts is given by using the mean diffusivity since the ratio of mean diffusivity to maximum diffusivity declines with an increasing diffusion anisotropy. Figure 7.2 shows the ventricles and peripheral fluid filled compartments now in red and the eyes in yellow. The major white matter...
Figure 7.1. Horizontal slices (with numbers) colour mapped with the maximum diffusivity.

Figure 7.2. Horizontal slices (with numbers) colour mapped with the mean diffusivity.

Figure 7.3. Horizontal slices (with numbers) colour mapped with the diffusion anisotropy.
7.2 Exploration of a Diffusion Tensor Data Set

Fiber tracts are still represented in saturated blue. The contrast between gray matter and white matter is now reduced since the diffusion in gray matter is isotropic.

This is confirmed in figure 7.3 which displays the diffusion anisotropy defined by equation 7.1. White matter is indicated by saturated blue with the corpus callosum and some parts of the internal capsule coloured red and yellow. Both gray matter and CSF have a very low anisotropy and are hence coloured in black-blue.

In the previous images anatomical structures were difficult to identify because of low contrast and because tissue regions were not differentiated from the background. An improvement is obtained by using exponential and cyclical colour maps as shown in figure 7.4. Part (a) of the figure visualizes the mean diffusivity on the left hand side with an exponential scale for the colour map (exponent of 2) and on the right hand side with a cyclical colour map (10 colour cycles). The ventricles and fluid filled compartments in the peripheral brain regions are now clearly identified by the red and yellow colours of the exponential map. The cyclical colour map displays additionally isocontours of constant mean diffusivity. The density of different isocontours indicates the gradient of the mean diffusivity. Note that outside of the fluid filled compartments the mean diffusivity is much more equally distributed.

Part (b) of the figure visualizes the diffusion anisotropy on the left hand side using an exponential colour map (exponent of 4) and on the right hand side using a cyclical colour map (10 colour cycles). The exponential colour map shows the major nerve fiber tracts around the lateral ventricle in red and yellow. The top and bottom structures are the genu (1) and the splenium (2) of the corpus callosum. The structures on the left and right hand side of the image are the genu (4), the anterior limb (5) and the posterior limb (3) of the internal capsule and the external capsule (6). The dark blue coloured region enclosed by the corpus callosum and the internal capsule represents the thalamus and the lateral ventricle. The yellow dot in its centre is the fornix (7). The internal capsule extends on the posterior side to form the optic radiation (8). The anisotropic regions in the periphery of the brain are due to eddy currents in CSF induced during DTI [PJB+96, JBP+98].

An interesting observation can be made from the image on the right hand side of figure 7.4 when comparing the isocontours in white matter regions in the left and right brain hemisphere. It can be seen that the anisotropy is almost symmetric in the anterior side of the brain but is slightly asymmetric in the posterior side with higher values for the left hemisphere. This result is in contrast to findings by [PGW+98] and could indicate that the white matter regions in the left hemisphere are more compact and therefore the fibers are more aligned [Fau00]. This hypothesis is consistent with the fact that the left hemisphere contains an additional brain region responsible for verbal abilities.

We believe that cyclical colour maps as demonstrated above could be especially useful when examining degenerative white matter diseases [LHM+99, HEM+99, TNBT+01].

Regions of gray matter, white matter, and CSF can be displayed simultaneously
Figure 7.4. Horizontal slice (number 20) with the mean diffusivity (a) and the diffusion anisotropy (b) visualized using an exponential (left) and a cyclical colour map (right).
7.2 Exploration of a Diffusion Tensor Data Set

Figure 7.5. Horizontal slice (number 20) coloured using a segmentation function (a) and a barycentric colour map (b). In (a) red, green and blue indicate white matter, CSF and grey matter, respectively.

using the segmentation function

\[
\lambda_{\text{segmentation}} = \begin{cases} 
1 & \text{if } \lambda_{\text{anisotropy}} \geq 0.25 \text{ and } 5 \times 10^{-6} < \lambda_{\text{mean}} < 1000 \times 10^{-6} \text{ mm/s} \\
2 & \text{if } \lambda_{\text{mean}} \geq 1000 \times 10^{-6} \text{ mm/s} \\
3 & \text{if } 5 \times 10^{-6} < \lambda_{\text{mean}} < 1000 \times 10^{-6} \text{ mm/s} \text{ and } \lambda_{\text{anisotropy}} < 0.25 \\
0 & \text{otherwise}
\end{cases}
\]

which is specified interactively using the toolkit’s expression field capabilities explained in section 5.3.

The conditions for the values 1, 2, and 3 are similar to the ones suggested by Pierpaoli et al. [PJB+96] and are chosen so that they indicate white matter, CSF and gray matter, respectively. Figure 7.5 (a) shows the resulting segmentation using the colours red, green and blue, respectively. In contrast to the previous images this image allows the identification of the thalamus as the two blue regions between the lateral ventricle in green and the internal capsule in red.

Regions of predominant linear anisotropic, planar anisotropic and isotropic diffusion are identified by using a barycentric colour map which visualizes the measures defined by equations 3.3–3.5.

A barycentric colour map is constructed by assigning three different colours to the vertices of a triangle and by interpolating these colours for each point \( P \) inside the triangle \( \Delta ABC \) using the barycentric coordinates

\[
\alpha = \frac{\text{area}(\Delta PBC)}{\text{area}(\Delta ABC)}, \quad \beta = \frac{\text{area}(\Delta PCA)}{\text{area}(\Delta ABC)}, \quad \gamma = \frac{\text{area}(\Delta PAB)}{\text{area}(\Delta ABC)}
\]

The barycentric coordinates define the weights of a convex sum of the triangle vertices which is equal to \( P \), i.e., \( \alpha A + \beta B + \gamma C = P \) where \( 0 \leq \alpha, \beta, \gamma \leq 1 \) and \( \alpha + \beta + \gamma = 1 \). The measures defined by equations 3.3–3.5 define a barycentric space of anisotropies and therefore can be visualized by mapping them onto the barycentric colour map.
Figure 7.5 (b) indicates that the diffusion is predominantly linear in the genu and splenium of the corpus callosum and in the internal capsule, and more planar in the optic radiation and in the more peripheral white matter regions. A higher linear anisotropy indicates a higher alignment of nerve fibers.

Figure 7.6. Horizontal slice (number 20) colour mapped with the oblate anisotropy (a) and the circular oblate anisotropy (b).

Figure 7.6 allows a more detailed investigation of this observation. Part (a) of the figure shows a horizontal map of the oblate anisotropy. The oblate anisotropy is highest in the splenium of the corpus callosum and the posterior limb of the internal capsule in the right hemisphere.

Part (b) of figure 7.6 displays the circular oblate anisotropy using an exponential colour map (exponent 3). The highest values, indicated in yellow, occur in the lateral regions of the splenium of the corpus callosum and might indicate the intersection with the posterior limb of the external capsule and the optic radiation. The relationship between a high circular anisotropy and the shape of diffusion ellipsoids is demonstrated by the enlargement on the right hand side of the figure. In areas of high circular oblate diffusion the ellipsoids become pizza-shaped.

7.2.2 Anatomical Landmarks

The identification of anatomical structures can be facilitated by inserting easily recognizable features into the 3D visualization. Two suitable structures are the ventricles and the eyes.

The ventricles are approximated by the $1700 \times 10^{-6} \text{mm/s}$-isosurface of the mean diffusivity. Since the brain is surrounded by CSF, isosurfaces are also computed for fluid filled fissures and sulci between the brain and the skull. These artifacts are subsequently removed using our surface cutting tool (page 172 of section 5.9). The resulting surface is shown in red in figure 7.7.

In order to allow a comparison with figure 3.17 the features in figure 7.7 are indicated with the same numbering: (1) cerebral aqueduct, (2) anterior horn, (4)
7.2 Exploration of a Diffusion Tensor Data Set

Figure 7.7. The eye balls (green) and the ventricles of the brain (red) represented as isosurfaces of the mean diffusivity.

posterior horn and (5) inferior horn of the lateral ventricle, (6) third ventricle and (7) fourth ventricle. The body of the lateral ventricle is similar in both figures whereas the fourth ventricle is larger in our image. The posterior and inferior horn of the lateral ventricle are less pronounced in our image which might be caused by a reduced diffusivity in the posterior and lateral horn since they lie outside the main flow direction of CFS [Guy87, p.34].

The eye balls provide another useful landmark. They are filled with a clear jelly (aqueous humor) and are consequently characterized by a very high isotropic diffusivity. Figure 7.7 shows the two eye balls defined as the $3500 \times 10^{-6} \text{mm/s}$-isosurface of the mean diffusivity coloured in green. The cavity in the front of each eye balls indicates the position of the lens. The dimensions of the left eye ball are $23.8\text{mm} \times 24.4\text{mm} \times 23\text{mm}$ which corresponds well with results reported in the literature which are an average diameter of $24\text{mm}$ and an average axial length of $25\text{mm}$ (including the cornea) [Wan].

Finally colour mapped surfaces can also be used as anatomical landmarks. A feature rich surface is given by the horizontal plane bisecting the eyes. Colour mapping it with the mean diffusivity indicates both the eyes and the ventricles and conveys to an experienced medical practitioner similar information as the two previously mentioned isosurfaces.
7.2.3 Introducing Local Tensor Information

Tensor Ellipsoids

Colour mapped surfaces only display scalar information. The information content can be improved by overlaying the surface with tensor ellipsoids introduced in subsection 5.9.3 which describe the diffusion direction.

Figure 7.8. Sagittal section colour mapped with the mean diffusivity and overlaid with diffusion ellipsoids at the DTI grid vertices: 1-genu of corpus callosum, 2-fornix, 3-splenium of corpus callosum, 4-lateral ventricle, 5-fourth ventricle.

Figure 7.8 shows a sagittal slice colour mapped with the mean diffusivity and overlaid with tensor ellipsoids at a regular sample grid. In order to avoid visual cluttering tensor ellipsoids are only drawn in regions containing nerve fibers. Such regions are characterized by a high mean diffusivity and a high diffusion anisotropy and can be detected by defining a filter with a mean diffusivity $\lambda_{\text{mean}} \leq 1250 \times 10^{-6} \text{mm/sec}$ and an anisotropy of $\lambda_{\text{anisotropy}} \geq 0.2$.

The image on the left hand side of figure 7.8 represents the section indicated by the white rectangle in the top image. The image on the right hand side shows the same section slightly rotate around its horizontal axis with its top edge tilted forward by about 45 degree. Fluid filled compartments show up as red, orange and yellow regions. The lateral ventricle (4) and the fourth ventricle (5) are clearly
identified. Fiber tracts are visualized by long thin ellipsoids. The image shows the fornix (2) and the corpus callosum which is the half-ring shaped area reaching from (1) to (3). Note that the fiber direction in the fornix is parallel to the image plane whereas the fibers in the corpus callosum are orthogonal to the image plane. Also note that the ellipsoids in the genu (1) and splenium (3) of the corpus callosum are extremely flat and that the anisotropy in these regions reaches a maximum. This effect is consistent with the horizontal maps of the anisotropy (figure 7.4 (b)) and the oblate anisotropy (figure 7.6).

Ellipsoid-based Textures

Informative textures can be created by colour mapping diffusion ellipsoids with directional information and/or anisotropy measures [Wün04a, WL04].

Subsection 4.4.3 explained that textures are perceptually characterized by the spatial frequency, contrast and orientation of texture components [Sch96, WK95]. In order to make use of the pattern recognition capabilities of the human visual system we use a dense distribution of colour mapped tensor ellipsoids and scale them such that they overlap in regions of high mean diffusivities. As a result regions of CSF exhibit an irregular pattern consisting of the visible sections of overlapping ellipsoids, white matter regions are characterized by regularly arranged long cigar shaped ellipsoids aligned along the nerve fiber tracts, and gray matter regions are represented by small approximately spherical ellipsoids. We encode additional information and further differentiate the textural appearance of regions with different tissue type by using two types of colour maps and by choosing black as a background colour.

White matter regions and the direction of nerve fiber tracts are emphasized by using the spherical colour map shown in figure 7.9. The hue, saturation and brightness of the colour spectrum vary along the circumferential, longitudinal and radial directions of the sphere, respectively. We encode the direction of the maximum diffusivity by computing its spherical coordinates and by associating them with the hue and saturation of the colour map. The diffusion anisotropy $\lambda_{anisotropy}$ is mapped onto the brightness parameter of the colour spectrum.

The results of applying this colour map to an ellipsoid-based texture are illustrated in figure 7.10. The visualization represents a horizontal slice through the brain. The colour mapped ellipsoids are illuminated using ambient, diffuse and specular illumination (top), ambient and diffuse illumination (middle) and ambient illumination only (bottom). The images on the right show a detail view of the images on the left.

It can be seen that specular illumination makes it difficult to differentiate tissue types since specular highlights dominate the image and obscure the ellipsoid colour and the texture pattern generated by overlapping/non-overlapping ellipsoids. Using ambient illumination only results in the best differentiation between tissue types: gray matter appears black, CSF appears as regions with large overlapping darkly coloured ellipsoids and white matter is represented by elongated ellipsoids with a highly saturated colour. However, using ambient illumination only makes it difficult to perceive the 3D shape of individual ellipsoids. Using ambient and diffuse illu-
Figure 7.9. A spherical colour map with hue, saturation and brightness varying along the circumferential, longitudinal and radial direction of the sphere, respectively. The illustration shows the surfaces of the colour map formed by choosing a constant brightness parameter of 1.0 (left) and 0.5 (right).

An alternative visualization is obtained by colouring diffusion ellipsoids with the barycentric colour map introduced in subsection 7.2.1. As a result regions of isotropic, planar anisotropic and linear anisotropic diffusion are represented by spherical blueish, flat reddish and elongated greenish ellipsoids, respectively. Figure 7.11 shows an example rendered with ambient illumination only (top-left) and with ambient, diffuse and specular illumination (top-right). Since the barycentric colour map does not use brightness and saturation variations both types of illumination result in effective visualizations with specular illumination being preferred if perception of the 3D ellipsoidal shape is important. A drawback of using the barycentric colour map is that it is difficult to differentiate gray matter regions from white matter regions with a diffusion perpendicular to the image plane.

Anisotropy Modulated Line Integral Convolution

Disadvantages of the methods introduced so far are that ellipsoids visualize the diffusion direction and hence the nerve fiber direction only at individual points, the 3D shape of ellipsoids can be hard to perceive, and there is little perceptual continuity between ellipsoids aligned along a nerve fiber tract.

As an improved method for visualizing the fiber direction in DTI slice images we propose Anisotropy Modulated Line Integral Convolution (AMLIC) [Wün04a]. A
Figure 7.10. Fiber tract direction over a horizontal slice through the brain visualized using an ellipsoid-based texture. The ellipsoids are colour mapped with the spherical colour map in figure 7.9 and illuminated using ambient, diffuse and specular illumination (top), ambient and diffuse illumination (middle) and ambient illumination only (bottom). The images on the right show an enlargement of the region containing the splenium of the corpus callosum.
Figure 7.11. Fiber tract direction over a horizontal slice through the brain visualized using an ellipsoid-based texture and the barycentric colour map shown in the bottom-left of the figure. The image in the top-left was rendered with ambient illumination only, the one in the top-right was rendered using ambient, diffuse and specular illumination.
2D AMLIC texture can be described in simple words as a blend of an LIC texture of the maximum diffusion direction with the colour mapped mean diffusivity.

In order to create an AMLIC texture we first compute the LIC texture from the maximum diffusion direction. The length of the convolution kernel at a pixel is proportional to the angle between the maximum diffusion direction and the textured surface. The LIC texture is colour mapped with the diffusion anisotropy. For each pixel a blending factor monotonically increasing with the diffusion anisotropy is defined. We obtained good results by using a power function with the exponent 0.2. The LIC texture is then blended with the colour mapped mean diffusivity using the OpenGL “GL BLEND” operation [WND97]. The texture colour of pixel \((i, j)\) is then given by:

\[
O_{ij} = \alpha_{ij}C_{ij} + (1 - \alpha_{ij})D_{ij} // \text{the output texture colour}
\]

where

\[
C_{ij} = L_{ij}c(\lambda_{\text{anisotropy}}(i, j)) // \text{the LIC texture colour mapped with the}
\]
\[
L_{ij} // \text{diffusion anisotropy at the centre of each pixel}
\]
\[
D_{ij} = d(\lambda_{\text{mean}}(i, j)) // \text{a texture representing the colour mapped mean diffusivity}
\]
\[
c(t), d(t) // \text{colour spectra indexed with } t \in [t_{\text{min}}, t_{\text{max}}]
\]
\[
\alpha_{ij} = (\lambda_{\text{anisotropy}}(i, j))^{0.2} // \text{a blending factor monotonically increasing with}
\]
\[
// \text{the diffusion anisotropy}
\]

The resulting texture has three properties: regions of high anisotropy feature a LIC like texture which indicates the 3D nerve fiber direction. Long texture elements indicate nerve fibers tangential to the textured surface whereas very short point-like texture elements indicate nerve fibers almost orthogonal to the textured surface. Regions which are not textured have a low anisotropy and therefore represent either gray matter or cerebral spinal fluid. If an appropriate colour map is chosen (we found that three equally distributed colours over the range of scalar values of the mean diffusivity work well) then gray matter is indicated by the colour(s) associated with low values of the mean diffusivity and CSF is indicated by the colour(s) associated with high values of the mean diffusivity.

Note that by blending the LIC texture and the colour mapped mean diffusivity we take into account the fact that there is no clear-cut boundary between white matter and gray matter. The resulting visualization therefore has advantages over image segmentation methods.

An example is given in figure 7.12. The cyan and green regions represent areas of high mean diffusivity and low anisotropy and therefore indicate fluid filled compartments. The whitish regions represent areas of low anisotropy and low mean diffusivity and therefore indicate gray matter. Textured regions exhibit a high anisotropy and indicate white matter. Very long texture components indicate fiber tracts parallel to the image plane, e.g., in the splenium of the corpus callosum (1). In contrast
Figure 7.12. Fiber tract direction over a horizontal slice through the brain visualized using anisotropy modulated line integral convolution (AMLIC). The nerve fibers in the splenium of the corpus callosum are parallel to the image plane (1) whereas nerve fibers in the posterior limb of the internal capsule (2) are almost vertical to it.

a noise-like texture with very short texture components indicates fiber tracts almost orthogonal to the image plane, e.g., in the posterior limb of the internal capsule (2).

### 7.2.4 Extracting Nerve Fiber Tracts

**Streamlines & Streamtubes**

As explained in subsection 3.3.4 nerve fiber tracts are characterized by a relatively high and anisotropic diffusion in the fiber direction. It is therefore possible to extract nerve fiber tracts as streamlines by integrating in the direction of the maximum diffusivity in regions of high anisotropy [Bas00, WL01b, WL04]. Since similar diffusion properties can also be registered due to noise or eddy currents in fluid filled compartments the maximum diffusivity at any step during the streamline integration must exceed a certain predefined limit and the streamline must exceed a specified minimum length.

Representing the tracked fibers by simple streamlines results in images where the exact 3D geometry is difficult to understand since depth cues due to shading and overlay are missing. A vastly improved visualization is obtained by using streamtubes which are formed by fitting a cylinder with constant radius around
Figure 7.13. Fiber tracts visualized using streamtubes colour mapped with the diffusion anisotropy. The image shows from left to right and top to bottom the superior, posterior, anterior, inferior and the left lateral side of the brain.
Case Study - The Diffusion Tensor Field in the Brain

the streamline. Figure 7.13 shows an example which was generated by using the start point condition $\lambda_{\text{anisotropy}} \geq 0.7$ and $\lambda_1 \geq 1000$ and the integration condition $\lambda_{\text{anisotropy}} \geq 0.3$ and $\lambda_1 \geq 750$. In order to obtain a dense image $2 \times 2 \times 2$ start points per grid cell of the DTI data set were chosen. Visual cluttering was avoided by limiting the maximum number of streamtubes intersecting a grid cell to eight. The step size used was $0.75mm$ and the minimum streamline length was $35mm$. Subsection 7.2.6 contains a discussion of the choice of parameters.

Hyperstreamlines

Streamlines and streamtubes contain only information about the major diffusion direction. More information can be visualized by using hyperstreamlines whose cross section represents the transverse diffusion in fiber tracts. If the hyperstreamlines’ diameters vary greatly this can lead to visual cluttering. Zhang et al. [ZCML00a, ZCML00b] therefore normalise the hyperstreamlines such that the maximum diameter is constant and refer to the resulting icons as streamtubes. Our results, obtained simultaneously with Zhang et al., suggest that this is unnecessary and leads to the loss of useful information.

Figure 7.14 depicts the nerve fiber tracts in the brain visualized with hyperstreamlines. The image shows the left lateral (a), posterior (b), superior (c) and anterior (d) side of the brain and a close-up view of the left lateral-posterior side (e). Perception of the 3D geometry is improved by inserting the eyes and the ventricles as anatomical landmarks as explained in subsection 7.2.2. The images were generated using one start point per grid cell of the DTI data set with a start point condition of $\lambda_{\text{anisotropy}} \geq 0.7$ and $\lambda_1 \geq 1000$ and an integration condition of $\lambda_{\text{anisotropy}} \geq 0.3$ and $\lambda_1 \geq 750$. Only one streamline per grid cell of the DTI data set was allowed and a step size of $0.75mm$ and a minimum length of $35mm$ were used for the computation. The hyperstreamlines are colour mapped with the maximum diffusivity.

Comparing the visualization with photographs and drawings from the literature [EW91, Guy87, Cyb, HWGR98, JB] makes it possible to identify the main fiber tracts introduced in subsection 3.3.1. The results were verified by consulting a neuroanatomist [Fau00].

Association fibers enable the communication between different areas of the cortex in the same hemisphere, and allow the integration of information from different parts of the cortex. Visible structures in figure 7.14 include the superior longitudinal fasciculus (6) which interconnects Broca’s area of motor speech with Wernicke’s area of language perception and the inferior occipito-frontal fasciculus (8) which connects the occipital and temporal lobes with the frontal lobes.

Commissural fibers cross over or join the two halves of the brain and enable communication between identical cortical areas in either hemisphere. The largest of these is the corpus callosum (2) which runs over the top of the lateral ventricles and has a genu and a splenium (7).

Projection fibers convey sensory information from the body to the cortex and motor information from the cortex down into the brainstem and spinal cord. The major projection system is the internal capsule (4) which is radially arranged where
Figure 7.14. Nerve fiber tracts visualized by hyperstreamlines colour mapped with the maximum diffusivity: 1-Corona radiata, 2-Corpus callosum, 3-Optic radiation, 4-Internal capsule, 5-Cerebral peduncles, 6-Superior longitudinal fasciculus, 7-Splenium of the corpus callosum, 8-Inferior occipito-frontal fasciculus. The image shows the left lateral (a), posterior (b), superior (c) and anterior (d) side of the brain and a close-up view of the left lateral-posterior side (e).
it leaves (or enters) the cortical mantle. The internal capsule continues as corona radiata (1) in superior direction and forms the cerebral peduncles (5) in the inferior direction where it is joined by the external capsule. The figure also shows the optic radiation (3) which is part of the visual pathway and forms the retrolenticular part of the internal capsule [Fau00]. The figure indicates that the optic radiation contains fibers from the internal capsule, the splenium of the corpus callosum and the superior longitudinal fasciculus.

Close to the eye balls three groups of hyperstreamlines can be differentiated and are indicated in figure 7.14 (d) by arrows. The arrow on the left denotes the optic nerve whereas the top and bottom arrow indicate the ophthalmic division and the maxillary division of the trigeminal nerve (cranial nerve V), which conducts sensory impulses from the cornea and the skin. The oculomotor nerve, which controls the eye muscle, is not visible presumably because of its relatively small diameter. The location of the optic nerve is more inward than expected from the literature [Guy87, p.41].

It is interesting to note that figure 7.14 shows groups of hyperstreamlines for each nerve even though the nerves have a relatively small diameter. This is most likely due to branching of the nerves and the presence of the ciliary ganglion. Note that the maximum diffusivity indicated by the colour mapping is considerably higher in the eye region than in all other white matter regions except for some parts of the corpus callosum.

In order to better differentiate features stricter conditions for the definition of hyperstreamlines can be used which results in fewer streamlines concentrated along the major nerve fiber tracts. Figure 7.15 shows a posterior, superior, and posterior left lateral view of a visualization obtained by using the same visualization parameters as in the previous image except that the starting condition for streamline integration was changed to $\lambda_{\text{anisotropy}} \geq 1.0$ and $\lambda_{1} \geq 1000$. The hyperstreamlines are colour mapped with the diffusion anisotropy.

The image indicates an approximately cylindrical anisotropy in most regions. Noticeable exceptions are the posterior limb of the internal capsule (4) and the splenium of the corpus callosum (5) where the minimum diffusivity is significantly reduced and the maximum diffusivity is increased which yields a high diffusion anisotropy.

Several interesting features can be identified from the figure. The hyperstreamline indicated by (2) inferior to the corpus callosum and bending downward is the fornix which takes part in the integrative function of the brain and terminates in the hippocampus. Superior to the corpus callosum is the cingulum (1) which connects the cingular jarus with the temporal lobe and the hippocampus. The pons is identified by the presence of the middle cerebellar peduncle (3) which wraps around the pyramidal tract.

The colour coding of the hyperstreamlines in figure 7.14 and 7.15 indicates that the mean diffusivity and the diffusion anisotropy usually don’t vary much even though the packing density of nerve fibers in different white matter regions can vary by up to a factor of five ($60000 - 70000/mm^2$ in the pyramidal tract and $338000/mm^2$ in the corpus callosum [PJB+96]).
Figure 7.15. Close up views of a visualization of fiber tracts using hyperstreamlines colour mapped with the diffusion anisotropy: 1-cingulum, 2-fornix, 3-middle cerebellar peduncle, 4-posterior limb of the internal capsule, 5-splenium of the corpus callosum.
7.2.5 Oblate Anisotropic Diffusion

In the previous subsection fiber tracts were identified by integrating along the principal diffusion direction in regions with a high diffusion anisotropy. Some regions, however, exhibit an oblate anisotropic diffusion [PJB+96], i.e., diffusion occurs primarily within a plane. Two major configurations can be distinguished: regions with $\lambda_1 \approx \lambda_2 >> \lambda_3$ exhibit a circular oblate anisotropy and are consistent with two orthogonal fiber tracts or white matter fibers randomly oriented on a plane [PJB+96]. Regions with $\lambda_1 > \lambda_2 >> \lambda_3$ exhibit a (non circular) oblate anisotropy and are consistent with the presence of fibers running in multiple direction but maintaining overall one principal direction [PJB+96]. The same effect can be caused by two non-perpendicular intersecting fiber tracts.

As a new visualization method we compute hyperstreamlines in regions with oblate anisotropy by integrating in the direction of the minimum diffusivity. At every point of the hyperstreamline the tangent then represents the normal of a surface exhibiting a predominantly planar diffusion. The shape of the cross section of the hyperstreamline describes the diffusion within the plane.

Figure 7.16. Hyperstreamlines in the minor eigenvector direction visualizing the oblate anisotropy (a) and the circular oblate anisotropy (b).
Figure 7.16 (a) depicts two views of a visualization of the oblate anisotropy. The image was produced using the start point condition $\lambda_{\text{o royalties}} \geq 0.5$ and $\lambda_1 \geq 800$ and the integration condition $\lambda_{\text{o royalties}} \geq 0.25$ and $\lambda_1 \geq 600$. For the integration a step size of 0.5mm and a minimum length of 17.5mm were chosen. The hyperstreamlines are colour coded with the oblate anisotropy. Hyperstreamlines in the peripheral regions of the brain were avoided by restricting the start points to a region of interest specified with the volume selection tool introduced in subsection 5.6.1.

Three major regions of oblate anisotropic diffusion can be identified. The region labelled “1” stretches from the beginning of the optic radiation towards the posterior limb of the internal capsule. The oblate diffusion within this region is probably due to fibers from the corpus callosum and the external capsule intersecting. Note that the cross section of the hyperstreamlines is very ellipsoidal which suggests that either the two fiber tracts intersect at an acute angle or that the water diffusion in one fiber tract is stronger.

The region labelled “2” is located close to the superior longitudinal fasciculus. Judging from the shape of the hyperstreamlines we suggest that this region represents the intersection of the superior longitudinal fasciculus with a commissural fiber tract, possibly an extension of the corpus callosum.

Finally the area around the pons, labelled “3”, also exhibits an oblate anisotropic diffusion. Note that the direction of the hyperstreamlines in the pons indicates an intersection of the cerebral peduncles with a laterally oriented fiber tract. The middle cerebellar peduncle is located outside the oblate anisotropic region.

Figure 7.16 (b) visualizes the circular oblate anisotropy and was generated using the same parameters as for the previous example except that the start point condition and the integration condition were given by $\lambda_{\text{circular oblate}} \geq 0.3$ and $\lambda_{\text{circular oblate}} \geq 0.2$, respectively. The image shows a subset of the previously identified regions, i.e., a circular anisotropic diffusion is found in some parts of the superior longitudinal fasciculus, the internal capsule and the pons.

### 7.2.6 Influence of parameters

Choosing the parameters for the streamline computation involves a compromise between the amount of information displayed, visual cluttering and the amount of noise and artifacts in the image. As mentioned previously noise and artifacts due to eddy currents can be minimised by specifying a suitable minimum length for the streamlines. Care must be taken that the specified value is not too high since otherwise short fiber tracts such as the ones in the vicinity of the eye are eliminated. We found that a minimum length of 35mm removed most of the short hyperstreamlines in the peripheral regions of the brain but had no noticeable influence on the anatomical features identified. Figure 7.17 shows two visualizations generated with a minimum length of 23mm (a) and 35mm (b).

Other important parameters are the choice of the start points for the computation and the integration condition along the trajectory of the hyperstreamline.
In the previously presented images the start points for fiber tracking were the centres of all grid cells of the DTI image data set which fulfill a given start point condition. The start point condition has to guarantee that streamlines are only generated inside of fiber tracts and we therefore specify both a minimum anisotropy and a minimum maximum diffusivity. The second restriction is necessary to eliminate “anisotropic noise”. The integration condition must guarantee that the streamline stays inside the fiber tract but has to allow for fluctuations in the anisotropy and maximum diffusivity due to noise. Consistency is best maintained by defining the integration condition as a weak form of the start point condition. Making the integration condition too strong results in streamlines with multiple gaps which often do not fulfill the minimum length criteria. Too weak a condition, on the other hand, generates few, if any, additional fiber tracts, but rather leads to visual cluttering.

\textbf{Figure 7.17.} Hyperstreamlines generated using the parameters in table 7.1.
Figure 7.17 shows as an example the images generated with the parameter choices in table 7.1. The hyperstreamlines are colour mapped with the maximum diffusivity. Part (b) and (c) of the figure demonstrate that increasing the minimum anisotropy at the start point from 0.6 to 1.0 removes the trigeminal nerve, most of the superior longitudinal fasciculus and the cingulum and optic nerve in the right hemisphere. Part (d) of the figure demonstrates that increasing the anisotropy instead to 0.7 and introducing additionally a minimum value for the maximum diffusivity maintains all previously identified fiber tracts but removes artifacts in the peripheral brain regions not eliminated by the minimum length condition. The nerve fiber tracts in the vicinity of the eye are most sensitive to variations of the start point and integration condition.

<table>
<thead>
<tr>
<th>Start point condition</th>
<th>Integration condition</th>
<th>min. length</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_{\text{anisotropy}} \geq 0.6$</td>
<td>$\lambda_{\text{anisotropy}} \geq 0.2$</td>
<td>23mm</td>
</tr>
<tr>
<td>$\lambda_{\text{anisotropy}} \geq 0.6$</td>
<td>$\lambda_{\text{anisotropy}} \geq 0.3$</td>
<td>35mm</td>
</tr>
<tr>
<td>$\lambda_{\text{anisotropy}} \geq 1.0$</td>
<td>$\lambda_{\text{anisotropy}} \geq 0.3$</td>
<td>35mm</td>
</tr>
<tr>
<td>$\lambda_{\text{anisotropy}} \geq 0.7$ and $\lambda_1 \geq 1000.0$</td>
<td>$\lambda_{\text{anisotropy}} \geq 0.3$ and $\lambda_1 \geq 1000.0$</td>
<td>35mm</td>
</tr>
</tbody>
</table>

Table 7.1. Parameter choices for the hyperstreamline images in figure 7.17.

7.3 Conclusion

Visualizing the diffusion tensor field in the brain gives an in vivo view of anatomical structures which was previously unobtainable. Using an incremental approach starting with colour mapped slices and extending to $2\frac{1}{2}$D and 3D techniques diffusion data can be systematically explored.

New insight into diffusion tensor data is gained using barycentric colour maps which show the distribution of anisotropies over a region and indicate possible fiber tract crossings. Cyclical colour maps reveal structure in scalar fields and by using them we found an interesting asymmetry in the diffusion anisotropy.

Ellipsoid-based textures provide a visualization of fiber direction and tissue types or anisotropy properties. A high amount of information can be encoded into slice images using anisotropy modulated line integral convolution. The technique does not only indicate three dimensional fiber direction but also provides a visual segmentation of tissue types.

Using streamtubes and hyperstreamlines makes it possible to obtain high quality 3D visualizations of the nerve fiber structure. We were able to demonstrate that the major principal pathways in the brain can be extracted and readily identified. Care must be taken when choosing the appropriate visualization parameters.

In contrast to Zhang et al. [ZCML00b] we do not normalise the transverse diffusivities encoded in hyperstreamlines and therefore we represent additional information. We also employ different methods for the selection of streamlines and we
perform the fiber tracking using a simple, fast and flexible general purpose streamline tracking algorithm implemented within our visualization toolkit.

The understanding of the complicated geometry of a fiber tract visualization can be improved by inserting anatomical landmarks. We listed several examples of landmarks and showed how they can be generated interactively with our toolkit. Advantages and disadvantages of the various methods were described.

New insight into the nerve fiber configuration was gained by applying hyper-streamlines in regions of oblate diffusion anisotropy which indicates planes of planar diffusion behaviour.

7.4 Future Research

In future we intend to use our visualization toolkit for the exploration of various white matter diseases. Current research indicates that DTI is superior to traditional MRI imaging and using our advanced visualization methods might help to better understand the development of various white matter diseases and the anatomical abnormalities in neurological disorders such as schizophrenia. Ideally we would like to obtain a series of data sets taken over time which makes it possible to create an animated visualization of the progress of a neurodegenerative disease.

We are also interested in qualitatively comparing our simple streamline tracking algorithms with specialised methods based on Markov-chain models and diffusion-advection processes.

Finally we want to visualize simultaneously a DTI data set and the corresponding functional MRI (fMRI) data set. By using visualization methods similar to the one suggested by Worsley et al. [WMN+96] it would be possible to simultaneously display anatomical and functional information. Functional MRI shows regions of activity in the brain due to the detection of metabolic processes characterized by a change in the blood oxygen level [Nyc99]. Improved very fast MR imaging sequences allow simultaneous fMRI and DTI [AKM+99]. This technique would be especially helpful when understanding regenerative processes in the brain for which little is known about the relationship between anatomical and functional regeneration [HG00].

Other functional brain imaging techniques have been presented that allow the measurement of electrophysiological, hemodynamic, and neurochemical processes that underlie normal and pathological brain function [BML01]. Initial work on the integration of MRI, fMRI, and EEG and MEG source reconstruction techniques is presented in [WF01] and we would like to further progress this exciting research.
Our thesis was motivated by the need of researchers to visualize tensor fields in biomedical finite element models. Commercial software packages proved to be limited in their capabilities for tensor field visualization, were not suitable for finite element models and often required extensive knowledge of computer graphic principles and scripting languages in order to use them appropriately. Various research groups contacting us for advice in visualizing their data sets confirmed this impression. Subsequently we got offered the task to visualize the diffusion tensor field in the brain. Diffusion tensor imaging is on the forefront of current research in neuroscience because of its promise to yield in vivo information about the neuroanatomy. The results have the potential to vastly improve the diagnoses and understanding of various neurodegenerative white matter diseases.

The goal of this thesis was therefore twofold. First we wanted to create an easy to use toolkit suitable for the visualization of tensor fields in biological tissue. Secondly we wanted to analyze the stress and strain field in the heart and the diffusion tensor field in the brain. This chapter reviews our achievements and lists our contributions to the fields of scientific visualization and biomedical imaging.

8.1 Achievements & Contributions

The main contribution of our research is a toolkit for visualizing biomedical finite element models. The toolkit comprises the standard visualization icons for scalar, vector and tensor fields, including colour maps, isosurfaces, height fields, vector glyphs, streamlines, line integral convolution, tensor ellipsoids, hyperstreamlines.

In addition the toolkit incorporates several novel features. The first feature is a modular design with separate objects describing input data sets, visualization settings, rendering parameters, and visualization windows. The design facilitates the definition of powerful simultaneous visualizations of multiple models such as the simultaneous display of a sick and a healthy heart. Using the same rendering
parameters ensures that both models are displayed using the same view, scaling, orientation and lighting.

The second feature is a generalized field structure. The user can define scalar fields, vector fields, and tensor fields by a FE mesh with interpolation functions, as a regular grid with a reconstruction filter or as analytic functions. New fields can be derived by using a set of predefined operators. The advantage of this construction is that fields only have to be created if they are really used and that the user can interactively construct new non-standard fields when important for a given application. Fields can also be represented with respect to both material and world coordinates.

The third novel feature of our toolkit is a global colour map control with an interface for designing new colour maps. New colour maps can be created by discretizing, exponentially stretching, or cyclically repeating a colour spectrum. Cyclic colour maps reveal isocontours and indicate gradient information without inducing visual cluttering and are useful when examining symmetry patterns and discontinuities in a scalar field. Colour map markers are a novel tool to indicate isocontours in a colour mapped domain. The width of the resulting isocontour indicates local gradient information. Another novel contribution is barycentric colour maps which are used to visualize three scalar fields simultaneously under the condition that the three fields form a barycentric coordinate system. The approach can be extended toward any number of scalar fields forming a convex sum.

As a new visualization tool we introduced filter objects. Filters can be used to define a domain for positioning visualization icons and to control a streamline integration. Filtering a point set for the placement of visualization icons introduces additional information into a visualization and reduces visual cluttering.

We also introduced an efficient and precise algorithm for computing length, area and volume measures. Regions of interest can be specified easily using a novel point, surface, and volume selection mechanism which is also used for the placement and combining of visualization icons.

In addition to the new tools explained above we implemented several useful improvements to existing icons such as a Marching Cubes algorithm for curvilinear FE meshes and a separation of tensor ellipsoids into six different coloured hexagonal sections to indicate the sign of the eigenvalues of the visualized tensors.

As a new visualization method for tensors we introduced anisotropy modulated line integral convolution (AM LIC). The technique convolves a noise texture with the major principal direction of the tensor and blends the texture with a colour map encoding the mean eigenvalue. The blending function is controlled by the ratio (anisotropy) of the eigenvalues. The direction of the major eigenvector relative to the image plane is encoded in the length of the convolution kernel.

An important issue in biomedical visualization is the perception of 3D structures. As a generalization of clipping planes we introduced a sectioning tool which allows the user to divide the model into 3D sections. Different sides of a model can be viewed simultaneously by placing mirrors into the scene. We showed that perception can be further improved by using markers, tags, raw image data and anatomical landmarks which are simple easily recognised features in a biomedical model.
8.1 Achievements & Contributions

An important requirement in biomedical imaging is a feature to communicate visualization results to laymen and experts at distant locations. An ideal medium to achieve this is the Internet. Our visualization package allows the user to save a visualization in Web3D (VRML) format which can be read using a standard web browser and VRML viewer which is available as a public domain browser plug-in. A visualization can be annotated using markers and name tags in order to identify features and regions of interest.

As a result of our research in the area of scientific visualization we presented a visualization schema which extends the traditional visualization pipeline by a visual interpretation step consisting of visual perception and cognition. Whereas the traditional approach represents only the encoding of data into visual attributes, visual interpretation represents a decoding of visual attributes. We also suggested a classification of visual attributes according to representational accuracy, perceptual dimension and spatial requirement. The classification supports finding suitable visual attributes for representing a given data set and hence forms the basis for mapping data onto visualization icons. As a further tool for supporting the visualization task we presented an extended classification of visualization icons by type, spatial domain, and information scope. We also provided a set of guidelines for selecting suitable visualization icons, for combining different visualization icons, and for increasing the effectiveness of a visualization.

This thesis concluded with two case studies presenting novel biomedical research. The first case study examined the deformation of the left ventricle of the human heart. The visualization of the healthy heart confirmed observations previously reported in the literature. Using tensor ellipsoids, streamlines and hyperstreamlines made it possible to visualize its complex deformation behaviour in a single image. Line integral convolution uncovered the presence of degenerate points at which the principal strains suddenly change direction. The visual information was supplemented by computing ventricular performance measures which are easily obtained from the finite element model using numerical integration. Colour mapping highlighted small continuity problems previously unknown to the developers of the ventricular model.

Visualizing a ventricle with dilated cardiomyopathy showed that the deformation of the lateral wall resembles most closely the expected motion whereas the septal wall behaved almost contrary to the expected deformation. Very large negative shear strains were recorded in the anterior-basal wall of the ventricle. The combined effect of these deformations seems to be a pumping action by shape deformation (from a circular to an ellipsoidal cross section) rather than by contraction.

The second case study introduced an incremental approach to explore the diffusion tensor field in a healthy brain. The use of various colour mapping techniques made it possible to identify several neuroanatomical structures. We also discovered an asymmetry in the diffusion anisotropy of the brain which could indicate differences in the alignment of white matter fibers. Furthermore we showed that slice images are improved by overlaying them with selected vector and tensor glyphs and that visual cluttering can be reduced by employing filters for the placing of the
New insight into diffusion tensor data is gained by using barycentric colour maps which depict the distribution of anisotropies over a region and indicate possible fiber tract crossings. A high amount of information can be encoded into a slice image by using ellipsoid-based textures or anisotropy modulated line integral convolution. Both techniques indicate the three dimensional fiber direction and provide a visual segmentation of tissue types. Anisotropy modulated line integral convolution has the added advantage that it gives a continuous representation of the fiber tract direction and a clearer differentiation of tissue types.

Three dimensional visualizations of the fiber tracts in the brain can be created by computing streamtubes or hyperstreamlines along the maximum principal diffusion direction. In contrast to Zhang et al. [ZCML00b] we do not normalise the transverse diffusivities represented in hyperstreamlines and therefore represent additional information. We also offer more freedom for the selection of streamlines and we perform the fiber tracking using a simple, fast and flexible general purpose streamline tracking algorithm implemented within our visualization toolkit. We identified most of the principal pathways in the brain and validated the visualization by consulting a neuroanatomist. The understanding of the complicated geometry of the fiber tract visualizations is improved by inserting anatomical landmarks which are interactively generated using our visualization package.

New insight into the nerve fiber configuration was gained by applying hyperstreamlines in the direction of the minimum principal diffusion in regions of oblate diffusion anisotropy and we were are able to identify several regions with a possible planar nerve fiber arrangement.
Our visualization toolkit is already extremely powerful but further research is needed to improve user-friendliness, interactivity, and the range of available features. Although the toolkit contains a finite element modelling system so far no user interface has been implemented for it. In future we would like to fully integrate the finite element modelling of biological systems with their visualization. We would also like to enhance the modular design of our package such that the same visualization parameters can be applied to different models in order to compare them more easily. This would require the notion of equivalence between meshes and fields defined on those meshes.

Interactivity of our toolkit could be improved by defining new element and point selection features to identify regions of interest and by improved interaction with the visualization icons such as dragging isosurfaces through the model domain. We are also interested in improving the interactive derivation of new field data sets. More research on the user interface design is needed to make the toolkit more accessible to medical experts (e.g., [Max00]). Ideally we would like to create a visual programming interface to derive new fields and to design visualizations.

In terms of supported visualization icons we would like to implement the 3D tensor topology and examine its relevance for interpreting 3D stress and strain fields. More research is currently underway examining the use of 3D LIC for tensor field visualization. Also of interest is the application of statistical methods and artificial intelligence (data mining) techniques to explore the relationship between different fields and between field components, and to analyse the structure of a field. Cluster analysis and neural networks are promising techniques in this context.

Tools supporting comparative visualization could prove useful in order to visualize differences between healthy and diseased tissue or in order to visualize the progress of a disease. Visualization of uncertainty might be appropriate when dealing with models where the reliability of data varies over the model domain.

Our case studies so far have concentrated on examining models of healthy organs. We are interested in analysing data sets of neurodegenerative and heart diseases.
Of special interest is the detection of ischemic myocardium. It is known that small changes in the deformation behaviour of the myocardium occur before first symptoms of a cardiac infarct develop and we hope that visualizing the stress and strain field gives an indication of regions of low blood perfusion. Visualization methods such as LIC and tensor topology might be especially promising for this purpose.

We are also interested in visualizing DTI data sets of various white matter diseases. Current research indicates that DTI is superior to traditional MRI imaging modalities and employing advanced visualization methods might improve the diagnosis and the understanding of the development of various neurodegenerative diseases such as schizophrenia and multiples sclerosis. An exciting project is the simultaneous visualization of DTI data and functional MRI data. Using visualization methods similar to the ones suggested by Worsley et al. [WMN+96] would make it possible to simultaneously display anatomical and functional information and to explore their relationship.

## 9.1 The Future of Medical Imaging

Cardiac MRI is predicted to become a comprehensive test of choice [POCD99]. An MRI toolkit for evaluating cardiac disease is progressing toward clinical reality and promises to have a major effect on the care of patients [BMM+98]. Its applications include the evaluation of vascular anatomy (coronary angiography, aortic disease, aberrant vessels, vascular access), cardiac anatomy (congenital anomalies, right ventricular dysplasia, constrictive pericarditis, valvular function), myocardial perfusion, myocardial wall motion [BMM+98] and visualization of metabolic processes using magnetic resonance spectroscopic imaging [SB99, p. 199]. Combining multiple MRI techniques in one examination replaces several other imaging procedures such as X-ray angiography, echocardiography, and scintigraphy and as such is cost effective and patient friendly [vR99].

Visualizing the strain field in the heart could yield even more insight if combined with the visualization of infarcted [vvd+91, PNP+99, LC99], ischemic and stunned myocardium [GKR+98, RJH+98] (using contrast MRI) and the visualization of the coronary artery tree (using MR coronary angiography) [RHH+97]. Of particular importance in future will be molecular imaging which can detect organic or cellular malfunction before first anatomical changes occur [LSHP03]. Further understanding might be obtained by visualizing and analysing the complex fluid dynamics of the blood flow in the heart [KYW+00].

A comprehensive future medical imaging and visualization system would communicate visualizations and diagnostic results using the internet [WK00, AG00] and might be connected to an expert system. Margulis and Sunshine [MS00] and Giger [Gig01] report that automatic diagnosis has already been achieved using neural networks and that applications such as MR treatment planning, computer assisted surgery and teleradiology are already becoming reality. Modern and future nuclear medicine workstations will network and allow incorporation into PACS with not only simultaneous viewing of US, CT, and MR images but also the option of image
9.2 Summary and Outlook

The field of medical imaging has increased rapidly in importance over the past decade. New image modalities and improved technologies allow the measurement of ever more data of the human body. Visualization packages like ours help transform this data into images which contain clearer and more accessible information. An ever closer alliance between medical imaging and scientific visualization will improve medical teaching and the understanding of the human body and its diseases. Visualizing medical data sets can enable general practitioners to perform a faster and more reliable diagnosis without consulting medical specialists. Complicated cases are already now frequently diagnosed over the internet using teleradiology [Hua01]. Much hope exists that this development will lead to faster and better treatment of patients, improved diagnostic, and lower health care costs. We are excited to participate in this development.
This thesis resulted in the following refereed international journal publications:


This thesis resulted in the following refereed international conference publications:


• The Visualization and Measurement of Left Ventricular Deformation, Burkhard Wünsche, Proceedings of the First Asia-Pacific Bioinformatics Conference, APBC ’03, 4-7 February 2003, Adelaide, Australia, ACS, pages 119-128.


The accompanying CD has the following content:

- The complete LaTeX source code of this thesis including all images and figures.
- The source code of our visualization toolkit and finite element modelling package.
- Several simple example data sets such as the plate with a hole under an uniaxial load. The medical image data sets used in the case studies are not included for privacy reasons.
- Several VRML files of visualizations of the brain data set.
C.1 Polar Decomposition of a Symmetric 2D Second-Order Tensor

While the eigenvalues and eigenvectors of a three-dimensional symmetric tensor are computed numerically it is possible to compute the eigenvalues and eigenvectors of a two-dimensional symmetric tensor exactly as shown below. We use this technique for the computation of the 2D tensor topology.

Let

\[ T = \begin{pmatrix} T_{11} & T_{12} \\ T_{12} & T_{22} \end{pmatrix} \]

be a symmetric 2D second-order tensor.

We want to compute the eigenvalues \( \lambda_{1,2} \) and eigenvectors \( \mathbf{e}_{1,2} \) of \( T \). The eigenvalues \( \lambda \) of \( T \) are the roots of the characteristic polynomial [Fis86]

\[
\det(T - \lambda I) = (T_{11} - \lambda)(T_{22} - \lambda) - T_{12}^2 = \lambda^2 - \lambda(T_{11} + T_{22}) + T_{11}T_{22} - T_{12}^2
\]

which are

\[
\lambda_{1,2} = \frac{T_{11} + T_{22}}{2} \pm \sqrt{\left(\frac{T_{11} + T_{22}}{2}\right)^2 - T_{11}T_{22} + T_{12}^2}
\]

Setting

\[
a = \frac{1}{2}(T_{11} + T_{22}) \\
b = \frac{1}{2}(T_{11} - T_{22}) \\
c = \sqrt{b^2 + T_{12}^2}
\]
we obtain \( \lambda_{1,2} = a \pm c \).

It remains to compute the eigenvectors. Note that an eigenvector multiplied with a scalar is again an eigenvector. Therefore it is sufficient to compute the angle \( \phi \in [-\frac{\pi}{2}, \frac{3\pi}{2}] \) of the eigenvector \( e_1 \) with the x-axis. Since \( e_2 \) must be orthogonal to \( e_1 \) we obtain

\[
e_1 = \begin{pmatrix} \cos \phi \\ \sin \phi \end{pmatrix} \quad e_2 = \begin{pmatrix} \sin \phi \\ -\cos \phi \end{pmatrix}
\]

Assuming that \( c \neq 0 \) we can use the fact that

\[
T = S^T \Lambda S
\]

where \( \Lambda = \text{diag}(\lambda_1, \lambda_2) \) and \( S = (e_1, e_2) \) [Fis86] and we obtain

\[
\begin{pmatrix} T_{11} & T_{12} \\ T_{12} & T_{22} \end{pmatrix} = \begin{pmatrix} \cos \phi & \sin \phi \\ \sin \phi & -\cos \phi \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} \cos \phi & \sin \phi \\ \sin \phi & -\cos \phi \end{pmatrix}
\]

\[
= \begin{pmatrix} \lambda_1 \cos^2 \phi + \lambda_2 \sin^2 \phi & \lambda_1 \sin \phi \cos \phi - \lambda_2 \sin \phi \cos \phi \\ \lambda_1 \sin \phi \cos \phi - \lambda_2 \sin \phi \cos \phi & \lambda_1 \sin^2 \phi + \lambda_2 \cos^2 \phi \end{pmatrix}
\]

This yields the equations

\[
T_{11} = \lambda_1 \cos^2 \phi + \lambda_2 \sin^2 \phi \tag{C.1}
\]

\[
T_{12} = (\lambda_1 - \lambda_2) \sin \phi \cos \phi \tag{C.2}
\]

\[
T_{22} = \lambda_1 \sin^2 \phi + \lambda_2 \cos^2 \phi \tag{C.3}
\]

Subtracting equation C.3 from equation C.1 gives

\[
(\lambda_1 - \lambda_2)(\cos^2 \phi - \sin^2 \phi) = T_{11} - T_{22} \tag{C.4}
\]

Using this result and equation C.2 we derive

\[
\frac{2T_{12}}{T_{11} - T_{22}} = \frac{2 \sin \phi \cos \phi}{\cos^2 \phi - \sin^2 \phi} = \frac{2 \sin \phi}{1 - \tan^2 \phi} = \frac{2 \tan \phi}{1 - \tan^2 \phi} = \tan 2\phi
\]

and hence

\[
\phi = \frac{1}{2} \arctan \left( \frac{T_{12}}{\frac{1}{2}(T_{11} - T_{22})} \right) = \frac{1}{2} \arctan \left( \frac{T_{12}}{b} \right)
\]

Note the usage of \( \arctan \left( \frac{x}{y} \right) = \arctan(y, x) \) which gives the arc tangent of \( \frac{x}{y} \) taking into account in which quadrant the point \((x, y)\) is, i.e., let \( z = \frac{x}{y} \) then

\[
\arctan \left( \frac{x}{y} \right) = \begin{cases} 
\arctan(z) & \text{if } y > 0 \\
\arctan(z) + \pi & \text{if } y < 0
\end{cases}
\]

In order to see that \( 2\phi = \arctan(b, T_{12}) \) gives the correct result consider equation C.4. We chose \( \lambda_1 = b + c \) and \( \lambda_2 = b - c \), hence \( \lambda_1 - \lambda_2 = 2c > 0 \). Also note
that $T_{11} - T_{22} = 2d$ and $\cos^2 \phi - \sin^2 \phi = 1 - 2\sin^2 \phi = \cos 2\phi$. We see that in order for equation C.4 to be true we require

$$\frac{-\pi}{2} \leq \phi \leq \frac{\pi}{2} \text{ if } d \geq 0 \text{ and } \frac{\pi}{2} \leq \phi \leq \frac{3\pi}{2} \text{ if } d < 0$$

which is fulfilled by the definition of \arctan(y, x).

\section*{C.2 Cylindrical Coordinates}

Section 2.2 introduced the concept of curvilinear coordinates. In this section we illustrate the concepts and principles presented previously by using as an example cylindrical coordinates. Cylindrical coordinates are popular in fluid dynamics and are therefore frequently used in bioengineering, e.g., for modelling blood flow.

Cylindrical coordinates $(r, \theta, z)$ are obtained from rectangular Cartesian (world) coordinates $(x, y, z)$ by using the transformation

$$r = \sqrt{x^2 + y^2}, \quad \theta = \arccos \frac{x}{\sqrt{x^2 + y^2}} \left( = \arcsin \frac{y}{\sqrt{x^2 + y^2}} \right), \quad z = z \quad (C.5)$$

where arccos is chosen such that a unique $\theta$ in $0 \leq \theta \leq 2\pi$ exists such that $\cos \theta = x/\sqrt{x^2 + y^2}$ and $\sin \theta = y/\sqrt{x^2 + y^2}$.

Figure C.1 shows an example of a point $P$ in world coordinates $(x, y, z)$ and cylindrical coordinates $(r, \theta, z)$. Using the notation from section 2.2 the Cartesian coordinates $(x, y, z)$ correspond to $(r_1, r_2, r_3)$ and the cylindrical coordinates $(r, \theta, z)$ correspond to $(q_1, q_2, q_3)$.

The Jacobian of the transformation C.5 is\footnote{Good online sources for differentiation formulas and other mathematical concepts are [Unia] and [Wei].}

$$J = \frac{\partial (r, \theta, z)}{\partial (x, y, z)} = \begin{pmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} & \frac{\partial r}{\partial z} \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial y} & \frac{\partial \theta}{\partial z} \\ \frac{\partial z}{\partial x} & \frac{\partial z}{\partial y} & \frac{\partial z}{\partial z} \end{pmatrix} = \begin{pmatrix} \frac{x}{\sqrt{x^2 + y^2}} & \frac{y}{\sqrt{x^2 + y^2}} & 0 \\ -\frac{y}{x^2 + y^2} & \frac{x}{x^2 + y^2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and its determinant is

$$|J| = \frac{1}{r}$$

Thus the inverse transformation exists at all points except of the origin and is given by

$$x = r \cos \theta, \quad y = r \sin \theta, \quad z = z$$
Figure C.1. A point $P$ in world coordinates $(x, y, z)$ and cylindrical coordinates $(r, \theta, z)$ together with its unitary basis $\{ \hat{r}, \hat{\theta}, \hat{z} \}$.

The Jacobian of the inverse transformation is

$$
\mathbf{J}^{-1} = \frac{\partial(x, y, z)}{\partial(r, \theta, z)} = \begin{pmatrix}
\frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} & \frac{\partial x}{\partial z} \\
\frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} & \frac{\partial y}{\partial z} \\
\frac{\partial z}{\partial r} & \frac{\partial z}{\partial \theta} & \frac{\partial z}{\partial z}
\end{pmatrix} = \begin{pmatrix}
\cos \theta & -r \sin \theta & 0 \\
\sin \theta & r \cos \theta & 0 \\
0 & 0 & 1
\end{pmatrix}
$$

It can be easily seen that $|\mathbf{J}^{-1}| = r$ and that the product of the two Jacobians is 1.

The coordinate curves of the cylindrical coordinate system are obtained by varying one coordinate and fixing the other two. The coordinate curves for $r$, $\theta$ and $z$ are

$$
\begin{pmatrix}
r \cos \theta_c \\
r \sin \theta_c \\
z_c
\end{pmatrix}, \quad \begin{pmatrix}
rc \cos \theta \\
r \sin \theta \\
z_c
\end{pmatrix}, \quad \text{and} \quad \begin{pmatrix}
rc \cos \theta_c \\
r \sin \theta_c \\
z
\end{pmatrix}
$$

and have a radial, circumferential and vertical direction, respectively. The subscript "c" indicates that the corresponding coordinate is fixed.

In order to determine the unitary basis for the cylindrical coordinate system consider a point $P = P(r, \theta, z)$. Using equation 2.7 the local basis vectors are computed as

$$
\hat{r} = \frac{\partial \mathbf{p}}{\partial r} = \frac{\partial x}{\partial r} \mathbf{e}_1 + \frac{\partial y}{\partial r} \mathbf{e}_2 + \frac{\partial z}{\partial r} \mathbf{e}_3 = \begin{pmatrix}
\cos \theta \\
\sin \theta \\
0
\end{pmatrix}
$$
\[ \hat{\theta} = \frac{\partial \mathbf{p}}{\partial \theta} = \frac{\partial x}{\partial \theta} \mathbf{e}_1 + \frac{\partial y}{\partial \theta} \mathbf{e}_2 + \frac{\partial z}{\partial \theta} \mathbf{e}_3 = \begin{pmatrix} -r \sin \theta \\ r \cos \theta \\ 0 \end{pmatrix} \]

\[ \hat{z} = \frac{\partial \mathbf{p}}{\partial z} = \frac{\partial x}{\partial z} \mathbf{e}_1 + \frac{\partial y}{\partial z} \mathbf{e}_2 + \frac{\partial z}{\partial z} \mathbf{e}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \]

and are illustrated in figure C.1.

### C.2.1 Coordinate Transformation of a Vector Quantity

![Diagram](image)

**Figure C.2.** A vector \( \mathbf{v} \) in world coordinates \((x, y, z)\) and cylindrical coordinates \((r, \theta, z)\).

Using equation 2.8 it is now possible to express a vector in both coordinate systems. For example, consider the vector

\[ \mathbf{v}_{\text{cylindrical coordinates}} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \]

in radial direction expressed in cylindrical coordinates. The same vector in world coordinates is

\[ \mathbf{v}_{\text{world coordinates}} = \mathbf{J}^{-1} \mathbf{v}_{\text{cylindrical coordinates}} = \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{x}{\sqrt{x^2 + y^2}} \\ \frac{y}{\sqrt{x^2 + y^2}} \\ 0 \end{pmatrix} \]

The two representations of the vector are illustrated in figure C.2.
Subsection 2.4.5 gave previously an overview of the FEM solution process. In summary the steps are:

**Step 1:** Choose element basis functions and discretize the domain.

**Step 2:** Create Galerkin Residual equations (obtained by rearranging the governing equations such that their right hand side is zero).

**Step 3:** Use the divergence theorem to create an integral equation involving the internal unknowns and the boundary normal derivatives (e.g., internal forces and external loads).

**Step 4:** Apply the finite element approximation to the integral equation and substitute the element trial solutions and weighting functions.

**Step 5:** Evaluate the element systems.

**Step 6:** Assemble the element systems to a global system.

**Step 7:** Insert the boundary conditions.

**Step 8:** Solve the global system.

**Step 9:** Compute the fluxes and other derived quantities.

The solution processes for FE problems usually only differ in the formulation of the Galerkin residual equations and the corresponding integral equations. Hence all steps save for step 2 and 3 proceed in a similar fashion. This appendix demonstrates the solution process using as examples the problems of 2D heat conduction and linear elasticity.
D.1 2D Heat Conduction

The heat conduction of an object with thermal conductivity \( k \) is described by the quasi-harmonic equation

\[-\nabla^T (k \nabla T) = Q\]

which in two dimensions is

\[-\frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) - \frac{\partial}{\partial y} \left( k \frac{\partial T}{\partial y} \right) = Q\]

where \( T \) is the temperature and \( Q \) the internally generated energy. Given a homogeneous 2D domain \( \Omega \) with boundary \( \Gamma \) and a given set of temperature and flux boundary condition the 2D heat conduction FE problem is now stated as finding the resulting steady-state temperature distribution \( T(x) \) over the domain \( \Omega \). The following computations use the vector form and apply therefore also to the equivalent 3D problems.

Step 1: Element Basis Functions and Discretization of the Domain

Figure D.1 shows the domain \( \Omega \) of a 2D heat conduction problem approximated using six nodes and two bilinear finite elements.

---

1The program on the accompanying CD contains a 2D and a 3D implementation of the slightly more general governing equation

\[-\frac{\partial}{\partial x} \left( \alpha_x \frac{\partial u(x,y)}{\partial x} \right) - \frac{\partial}{\partial y} \left( \alpha_y \frac{\partial u(x,y)}{\partial y} \right) + \beta u(x,y) = h(x,y)\]

or in vector form

\[-\nabla^T (\alpha \nabla u) + \beta u = h\]

The equation is sometimes referred to as the Sturm-Liouville boundary problem and is also used to describe 2D electrostatic fields, 2D flows of incompressible inviscid fluid, and 3D convection problems [Bur87].
Step 2: Create Galerkin Residual Equations

In order to keep with our general notation for the unknown variable we denote the temperature distribution $T$ with $u(x)$ and the internally generated heat $Q$ with $h(x)$. The governing equation is rewritten by replacing $u$ with the FE approximation function $\tilde{u}$, i.e.,

$$
-\frac{\partial}{\partial x} \left( k \frac{\partial \tilde{u}}{\partial x} \right) - \frac{\partial}{\partial y} \left( k \frac{\partial \tilde{u}}{\partial y} \right) = h
$$

(D.3)

and in vector form

$$
-\nabla^T (k \nabla \tilde{u}) = h
$$

(D.4)

Using this equation the Galerkin residual, which is a measure of the error in our solution, is given by

$$
R = R(x) = -\nabla^T (k \nabla \tilde{u}) - h
$$

(D.5)

The residual is weighted using a weighting function $\omega = \omega(x)$ and is integrated over the domain, to obtain a total residual which, ideally, should be zero, i.e.,

$$
\int_{\Omega} R \omega d\Omega = 0
$$

(D.6)

By substituting equation D.5 this equation becomes

$$
\int_{\Omega} \left( -\nabla^T (k \nabla \tilde{u}) - h \right) \omega d\Omega = 0
$$

(D.7)

For the Galerkin FEM the weighting function will be replaced in the FE approximation step (step 4) with each of the basis functions in turn, yielding a set of equations. Solving that set of equations yields a solution for $\tilde{u}(x)$ in which the error is minimal in an integral sense with respect to all trial functions of the form D.10 [BP91, Pro96].

Step 3: Divergence Theorem

Using the product rule

$$
\nabla^T (\nabla vw) = \nabla^T \nabla vw + (\nabla v)^T \nabla w
$$

rearranging it

$$
-\nabla^T \nabla vw = -\nabla^T (\nabla vw) + (\nabla v)^T \nabla w
$$

and applying it to equation D.7 yields

$$
- \int_{\Omega} \nabla^T \left( k \nabla \tilde{u} \omega \right) d\Omega + \int_{\Omega} \left( (k \nabla \tilde{u})^T \nabla \omega - h \omega \right) d\Omega = 0
$$

(D.8)

The first integral is transformed by the Gauss-Green theorem (divergence theorem, e.g., [Heu81]) as

$$
- \int_{\Omega} \nabla^T \left( k \nabla \tilde{u} \omega \right) d\Omega = \oint_{\Gamma} (-k \nabla \tilde{u} \omega)^T nd\Gamma
$$
\[
\begin{align*}
&= \oint_{\Gamma} (-k \nabla \tilde{u})^T n \omega d\Gamma \\
&= \oint_{\Gamma} \tau^T n \omega d\Gamma \\
&= \oint_{\Gamma} \tau_n \omega d\Gamma \\
&= -\oint_{\Gamma} \tau_{-n} \omega d\Gamma
\end{align*}
\]
so that equation D.8 becomes
\[
\int_{\Omega} (k \nabla \tilde{u})^T \nabla \omega = \int_{\Omega} h \omega d\Omega + \oint_{\Gamma} \tau_{-n} \omega d\Gamma \tag{D.9}
\]
Here \( n = (n_x, n_y)^T \) is the outward normal of the element boundary \( \Gamma \), \( \oint d\Gamma \) is the integral around the element boundary in counter-clockwise direction, \( \tau = -k \nabla \tilde{u} \) is the flux, and \( \tau_n = \tau_x n_x + \tau_y n_y \) is the boundary flux in the normal direction. Since the heat flux is usually specified as going into the domain (i.e., energy is added) the inward normal \(-n\) is used.

**Step 4: Substitute Element Trial Solutions and Weighting Functions**

For each element \( e \) the residual equation D.9 has to be fulfilled by an *element trial solution* \( \tilde{u}^{(e)} \) of the form
\[
\tilde{u}^{(e)} = \tilde{u}^{(e)}(x) = \sum_{j=1}^n u_j^{(e)} \phi_j(x) \tag{D.10}
\]
where \( \phi_j \) are the element basis functions (see subsection 2.4.2).

The element equations of the Galerkin FEM are then formed by substituting the element trial solution D.10 into the residual equation D.9 and by replacing the weighting function with each element basis function, i.e.,
\[
\begin{align*}
\sum_{j=1}^n \left( \int \int \left[ k \frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} + k \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} \right] dx dy \right) u_j^{(e)} &= \int \int h \phi_i dx dy + \oint \tau_{-n}^{(e)} \phi_i ds \\
&= \int \int h \phi_i dx dy + \oint \tau_{-n}^{(e)} \phi_i ds \tag{D.11} \quad i = 1, \ldots, n
\end{align*}
\]
or in matrix form
\[
K^{(e)} u^{(e)} = f^{(e)} \tag{D.12}
\]
where \( K^{(e)} \) is the *element stiffness matrix* with components
\[
K_{ij}^{(e)} = \int \int \left( k \frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} + k \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} \right) dx dy \tag{D.13}
\]
\( u^{(e)} = (u_1^{(e)}, \ldots, u_n^{(e)})^T \) is the solution vector, and \( f^{(e)} \) is the *element load vector* with the components
\[
f_i^{(e)} = \int \int h \phi_i dx dy + \oint \tau_{-n}^{(e)} \phi_i ds \tag{D.14}
\]
Step 5: Evaluate Element Systems

In order to evaluate equations D.13–D.14 they are rewritten in terms of the material coordinates \(\xi = (\xi, \mu)\) using the substitution rule for multidimensional integration ([Heu81])

\[
K_{ij}^{(e)} = \int_0^1 \int_0^1 \left( k \frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} + k \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} \right) |\det \mathbf{J}^{(e)}| d\xi d\mu \tag{D.15}
\]

\[
f_i^{(e)} = \int_0^1 h \phi_i |\det \mathbf{J}^{(e)}| d\xi d\mu + \oint_{\mathcal{T}^{(e)}_n} \phi_i ds = f_{1,i}^{(e)} + f_{2,i}^{(e)} \tag{D.16}
\]

where the Jacobian \(\mathbf{J}^{(e)}\) and the partial derivatives with respect to the world coordinates are computed as shown in subsection 2.4.3.

The two dimensional integrals are evaluated by gauss quadrature

\[
K_{ij}^{(e)} = \sum_{k=1}^m \sum_{l=1}^m w_k w_l \left[ \left( k \frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} + k \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} \right) |\det \mathbf{J}^{(e)}| \right]_{(\xi_k, \mu_l)} \tag{D.17}
\]

\[
f_{1,i}^{(e)} = \sum_{k=1}^m \sum_{l=1}^m w_k w_l \left[ h \phi_i |\det \mathbf{J}^{(e)}| \right]_{(\xi_k, \mu_l)} \tag{D.18}
\]

Since bilinear elements and a second degree governing equation was used \(m = 2\) gauss points in each dimension are sufficient ([Bur87, page 617]). The gauss points are

\[
\xi_1 = \mu_1 = \frac{1}{2} \left( 1 - \frac{1}{\sqrt{3}} \right), \quad \xi_2 = \mu_2 = \frac{1}{2} \left( 1 + \frac{1}{\sqrt{3}} \right)
\]

It remains to evaluate the boundary flux integral \(f_{2,i}^{(e)}\) in equation D.16. Burnett shows [Bur87, page 617ff.] that it is sufficient to consider only element boundaries that are domain boundaries. The flux boundary integral therefore represents the addition of the natural boundary conditions to the global system and is introduced in step 7.

Step 6: Assembly of the Global System

The element systems as given by equation D.12 are assembled into a global system

\[
\mathbf{K} \mathbf{u} = \mathbf{f} \tag{D.19}
\]

by inserting the entries of the element stiffness matrices \(\mathbf{K}^{(e)}\) and element load vectors \(\mathbf{f}^{(e)}\) into the global stiffness matrix \(\mathbf{K}\) and into the global load vector \(\mathbf{f}\), respectively. If a node is shared by several elements each element contributes a term to the corresponding position in the system and the terms are added together.

To demonstrate the process consider figure D.1. The global nodes 1, 2, 4, and 5 are the local nodes 1-4 of element \(e_1\) and the global nodes 2, 3, 5, and 6 are the local
nodes 1-4 of element $e_2$. The element equations for element 1 and 2 then have the form

$$
\begin{pmatrix}
K_{11}^{(e_1)} & K_{12}^{(e_1)} & K_{13}^{(e_1)} & K_{14}^{(e_1)} \\
K_{21}^{(e_1)} & K_{22}^{(e_1)} & K_{23}^{(e_1)} & K_{24}^{(e_1)} \\
K_{31}^{(e_1)} & K_{32}^{(e_1)} & K_{33}^{(e_1)} & K_{34}^{(e_1)} \\
K_{41}^{(e_1)} & K_{42}^{(e_1)} & K_{43}^{(e_1)} & K_{44}^{(e_1)}
\end{pmatrix}
\begin{pmatrix}
u_1^{(e_1)} \\
u_2^{(e_1)} \\
u_3^{(e_1)} \\
u_4^{(e_1)}
\end{pmatrix}
= 
\begin{pmatrix}
f_1^{(e_1)} \\
f_2^{(e_1)} \\
f_3^{(e_1)} \\
f_4^{(e_1)}
\end{pmatrix}
$$

Assembling these two systems into a global system yields

$$
\begin{pmatrix}
K_{11}^{(e_1)} & K_{12}^{(e_1)} & K_{13}^{(e_1)} & K_{14}^{(e_1)} & K_{15}^{(e_2)} & K_{16}^{(e_2)} & K_{17}^{(e_2)} & K_{18}^{(e_2)} \\
K_{21}^{(e_1)} & K_{22}^{(e_1)} & K_{23}^{(e_1)} & K_{24}^{(e_1)} & K_{25}^{(e_2)} & K_{26}^{(e_2)} & K_{27}^{(e_2)} & K_{28}^{(e_2)} \\
K_{31}^{(e_1)} & K_{32}^{(e_1)} & K_{33}^{(e_1)} & K_{34}^{(e_1)} & K_{35}^{(e_2)} & K_{36}^{(e_2)} & K_{37}^{(e_2)} & K_{38}^{(e_2)} \\
K_{41}^{(e_1)} & K_{42}^{(e_1)} & K_{43}^{(e_1)} & K_{44}^{(e_1)} & K_{45}^{(e_2)} & K_{46}^{(e_2)} & K_{47}^{(e_2)} & K_{48}^{(e_2)} \\
K_{51}^{(e_2)} & K_{52}^{(e_2)} & K_{53}^{(e_2)} & K_{54}^{(e_2)} & K_{55}^{(e_2)} & K_{56}^{(e_2)} & K_{57}^{(e_2)} & K_{58}^{(e_2)} \\
K_{61}^{(e_2)} & K_{62}^{(e_2)} & K_{63}^{(e_2)} & K_{64}^{(e_2)} & K_{65}^{(e_2)} & K_{66}^{(e_2)} & K_{67}^{(e_2)} & K_{68}^{(e_2)} \\
K_{71}^{(e_2)} & K_{72}^{(e_2)} & K_{73}^{(e_2)} & K_{74}^{(e_2)} & K_{75}^{(e_2)} & K_{76}^{(e_2)} & K_{77}^{(e_2)} & K_{78}^{(e_2)} \\
K_{81}^{(e_2)} & K_{82}^{(e_2)} & K_{83}^{(e_2)} & K_{84}^{(e_2)} & K_{85}^{(e_2)} & K_{86}^{(e_2)} & K_{87}^{(e_2)} & K_{88}^{(e_2)}
\end{pmatrix}
\begin{pmatrix}
u_1 \\
u_2 \\
u_3 \\
u_4 \\
u_5 \\
u_6 \\
u_7 \\
u_8
\end{pmatrix}
= 
\begin{pmatrix}
f_1^{(e_1)} + f_1^{(e_2)} \\
f_2^{(e_1)} + f_2^{(e_2)} \\
f_3^{(e_1)} + f_3^{(e_2)} \\
f_4^{(e_1)} + f_4^{(e_2)} \\
f_5^{(e_2)} \\
f_6^{(e_2)} \\
f_7^{(e_2)} \\
f_8^{(e_2)}
\end{pmatrix}
$$

Step 7: Inserting Boundary Conditions

The governing equation D.3 of the heat conduction problem is a partial differential equation of degree two and therefore contains two types of boundary conditions (BCs): Essential boundary conditions specify $\hat{u}(x)$ along the boundary and natural boundary conditions specify the flux $\tau_n(x)$ normal to the boundary, where

$$
\tau_n = -\mathbf{n}^T \mathbf{f} = k \frac{\partial u}{\partial x} n_x + k \frac{\partial u}{\partial y} n_y
$$

In order to get a unique solution one boundary condition must be specified for each boundary node. At least one essential BC must be specified since otherwise the solution is indeterminate with respect to a constant additive term.

It remains to apply the natural and essential BC conditions to the global system, i.e., the global system must be modified in a way such that the solution fulfills the given boundary conditions. Natural boundary conditions are given as point fluxes and can be directly inserted into the load vector as $f_i = \tau_{n,i}$.

Essential boundary conditions are applied by inserting them directly into the solution vector. If an essential boundary condition $\hat{u}$ is given for node $k$ then the solution vector must fulfill $u_k = \hat{u}$. To achieve this set $K_{kk}(e_1, e_2) = 1$ and $K_{ik} = 0, \ i = 1, \ldots, n, \ i \neq k$ in the global stiffness matrix and set $f_k = \hat{u}$ in the global load vector. In order to keep the global stiffness matrix symmetric set $K_{ik} = 0, \ i = 1, \ldots, n, \ i \neq k$ and subtract $\hat{u}K_{ik}$ from $f_i$ for $i = 1, \ldots, n, \ i \neq k$.

Again the process is best demonstrated by an example. Below we show the modifications to a global system with three nodes and the essential boundary condition $u_2 = \hat{u}$.

$$
\begin{pmatrix}
K_{1,1} & K_{1,2} & K_{1,3} \\
K_{2,1} & K_{2,2} & K_{2,3} \\
K_{3,1} & K_{3,2} & K_{3,3}
\end{pmatrix}
\begin{pmatrix}
u_1 \\
u_2 \\
u_3
\end{pmatrix}
= 
\begin{pmatrix}
f_1 \\
f_2 \\
f_3
\end{pmatrix}
$$
D.1 2D Heat Conduction

\[
\begin{pmatrix}
K_{11} & K_{12} & K_{13} \\
0 & 1 & 0 \\
K_{31} & K_{32} & K_{33}
\end{pmatrix}
\begin{pmatrix}
u_1 \\
u_2 \\
u_3
\end{pmatrix}
= 
\begin{pmatrix}
f_1 \\
\hat{u} \\
f_3
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
K_{1,1} & K_{1,3} \\
0 & 1 & 0 \\
K_{3,1} & K_{3,3}
\end{pmatrix}
\begin{pmatrix}
u_1 \\
u_2 \\
u_3
\end{pmatrix}
= 
\begin{pmatrix}
f_1 - \hat{u}K_{12} \\
\hat{u} \\
f_3 - \hat{u}K_{32}
\end{pmatrix}
\]

Step 8: Solving the Global System

Since basis functions are local to elements the global system of the FEM is usually sparse, i.e., most of the matrix elements are zero. Furthermore, the global matrix usually has a band structure (depending on the chosen numbering of nodes) so that the global system can be efficiently solved using band matrix solvers [Bur87]. Alternative methods employed in FE problems include the preconditioned conjugate gradient method, Cholesky decomposition and iterative methods such as the Gauss-Seidel iteration [BP91].

Step 9: Computing Fluxes

In many applications, the boundary nodal fluxes resulting from the solution to the global system are of special interest. The easiest way to compute them is as

\[
\tilde{\mathbf{f}} = \mathbf{Ku}
\]  

(D.21)

where \( \mathbf{u} \) is the solution vector resulting from solving the global system after insertion of the boundary conditions and \( \mathbf{K} \) is the global stiffness matrix before insertion of the boundary conditions as in equation D.19. The boundary nodal fluxes are then the components of \( \tilde{\mathbf{f}} \) corresponding to boundary nodes, i.e., \( \tau_{-n,i} = f_i \) if \( i \) is the index of a boundary node [HP02]. An alternative way to compute the normal fluxes is by averaging the element fluxes [Bur87, page 620].
The Finite Element Method - Examples

D.2 Linear Elasticity

The theory of linear elasticity is used to predict the deformation of a (nearly) rigid body under applied loads. An introduction to the topic was given in chapter 2. This section explains the steps 2-4 of the FE modelling of a linear elastic solid. The other solution steps are analogous to the ones for the 2D heat conduction problem (see section D.1).

Step 2: Create Galerkin Residual Equations

The governing equation for the problem of a $d$-dimensional linear elastic solid under an applied load was given by equation 2.12 as

$$\sum_{j=1}^{d} \frac{\partial \sigma_{ij}}{\partial x_j} + f_i = 0 \quad i=1,\ldots,d$$

or in matrix form

$$\nabla^T \sigma + f = 0$$

where, for $d=3$, $u^T = (u_x, u_y, u_z)$ is the displacement vector, $f^T = (f_x, f_y, f_z)$ is the internal load vector,

$$\sigma^T = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix}$$

is the stress vector and

$$\nabla^T = \begin{pmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \end{pmatrix}$$

is the generalized divergence operator.

The residual $R$ is then defined as

$$R = R(x) = \nabla^T \sigma + f = 0$$

which multiplied by a weighting function and integrated gives the residual function

$$\int_{\Omega} R \omega d\Omega = \int_{\Omega} (\nabla^T \sigma + f) \omega d\Omega = 0 \quad (D.22)$$

Step 3: Divergence Theorem

Rearranging terms and applying the product rule yields

$$\int_{\Omega} \nabla^T \omega \sigma d\Omega = \int_{\Omega} \nabla^T (\sigma \omega) d\Omega + \int_{\Omega} f \omega d\Omega$$

Using the following generalization of the divergence theorem\(^2\)

$$\int_{\Omega} \nabla^T (\sigma \omega) d\Omega = \int_{\Gamma} (\sigma \omega)^T n d\Gamma \quad (D.23)$$

\(^2\)A proof of equation D.23 is given in section D.3.
with the matrix form of $\sigma$ on the right-hand side gives
\[
\int_\Omega \nabla^T \omega \sigma d\Omega = \int_\Gamma (\sigma \omega)^T n d\Gamma + \int_\Omega f \omega d\Omega
\]
\[
= \int_\Gamma \sigma^T n \omega d\Gamma + \int_\Omega f \omega d\Omega
\]
\[
= \int_\Gamma \tau_n \omega d\Gamma + \int_\Omega f \omega d\Omega
\]
where $\Gamma$ is the boundary of the domain $\Omega$, $n$ is its normal, $\nabla$ is the generalized gradient operator, which is adjoint to the generalized divergence operator $\nabla^T$, and $\tau_n$ is its normal, i.e.,
\[
\tau_n = \sigma^T n = \begin{pmatrix}
\sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\
\sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\
\sigma_{zx} & \sigma_{zy} & \sigma_{zz}
\end{pmatrix}
\begin{pmatrix}
x \\
y \\
z
\end{pmatrix} = \begin{pmatrix}
\sigma_{xx} n_x + \sigma_{xy} n_y + \sigma_{xz} n_z \\
\sigma_{yx} n_x + \sigma_{yy} n_y + \sigma_{yz} n_z \\
\sigma_{zx} n_x + \sigma_{zy} n_y + \sigma_{zz} n_z
\end{pmatrix}
\]
is the boundary normal stress or traction.

**Step 4: Substitute Element Trial Solutions**

For each element the solution for the unknown displacement field has the form
\[
\mathbf{u}^{(e)}(x) = \sum_{j=1}^{n} \mathbf{u}_j^{(e)} \phi_j
\]
The strain vector hence is
\[
e^{(e)} = \begin{pmatrix}
\frac{\partial \phi_1}{\partial x} \\
\frac{\partial \phi_1}{\partial y} \\
\frac{\partial \phi_1}{\partial z} \\
\vdots \\
\frac{\partial \phi_n}{\partial x} \\
\frac{\partial \phi_n}{\partial y} \\
\frac{\partial \phi_n}{\partial z}
\end{pmatrix}
\begin{pmatrix}
0 & 0 & \ldots & 0 & 0 \\
0 & 0 & \ldots & 0 & 0 \\
0 & 0 & \ldots & 0 & 0 \\
1/2 \frac{\partial \phi_1}{\partial y} & 1/2 \frac{\partial \phi_1}{\partial x} & 0 & 1/2 \frac{\partial \phi_1}{\partial z} & 0 \\
1/2 \frac{\partial \phi_1}{\partial y} & 1/2 \frac{\partial \phi_1}{\partial x} & 0 & 1/2 \frac{\partial \phi_1}{\partial z} & 0 \\
0 & 0 & \ldots & 0 & 0 \frac{\partial \phi_n}{\partial x} & \frac{\partial \phi_n}{\partial y} & \frac{\partial \phi_n}{\partial z}
\end{pmatrix}
\begin{pmatrix}
\mathbf{u}_1^{(e)} \\
\mathbf{u}_2^{(e)} \\
\vdots \\
\mathbf{u}_n^{(e)} \\
\mathbf{u}_1^{(e)}
\end{pmatrix}
\]
and substituting the expression into the constitutive relation 2.15 yields
\[
\sigma^{(e)} = CB\mathbf{u}^{(e)}
\]
\[
(D.26)
\]
\[\text{Note that if the material is inhomogeneous the constitutive matrix } C \text{ is not constant, i.e., } C = C(x) = C^{(e)}(\xi).\]
Rewriting equation D.24 \( n \) times with the weighting function replaced by the basis functions \( \phi_i \) gives

\[
\int_{\Omega^{(e)}} B^T \sigma^{(e)} d\Omega = \int_{\Gamma^{(e)}} \Phi^T \tau_n^{(e)} d\Gamma + \int_{\Omega^{(e)}} \Phi^T \Gamma^{(e)} d\Omega
\]

where

\[
\Phi = \begin{pmatrix}
\phi_1 & 0 & 0 & \cdots & \phi_n & 0 & 0 \\
0 & \phi_1 & 0 & \cdots & \phi_n & 0 & 0 \\
0 & 0 & \phi_1 & \cdots & \phi_n & 0 & 0
\end{pmatrix}
\]

and by inserting equation D.26

\[
\int_{\Omega^{(e)}} B^T C B u^{(e)} d\Omega = \int_{\Gamma^{(e)}} \Phi^T \tau_n^{(e)} d\Gamma + \int_{\Omega^{(e)}} \Phi^T \Gamma^{(e)} d\Omega
\]

The resulting \( 3n \) equations can be written in the usual condensed matrix form

\[
K^{(e)} u^{(e)} = b^{(e)}
\]

where

\[
K^{(e)} = \int_{\Omega^{(e)}} B^T C B u^{(e)} d\Omega
\]

is the element stiffness matrix and

\[
b^{(e)} = b_f^{(e)} + b_r^{(e)} = \int_{\Omega^{(e)}} \Phi^T \tau_n^{(e)} d\Gamma + \int_{\Omega^{(e)}} \Phi^T \Gamma^{(e)} d\Omega
\]

is the element load vector consisting of the external loads \( b_f^{(e)} \) and the internal loads \( b_r^{(e)} \).

Note that the resulting element systems have the same form as equation D.12 in the previous example and can therefore be evaluated and assembled analogously. The essential boundary conditions are now displacement vectors and the natural boundary conditions are surface traction vectors. The global system can be solved resulting into the unknown displacement values at the mesh nodes. Stresses and strain are computed by evaluating equations D.25 and D.26, respectively, using the solution vector for the unknown displacement.


\section*{D.3 Divergence Theorem for a Symmetric Tensor}

Claim:

\[ \int_{\Omega} \nabla^T (\sigma \omega) d\Omega = \int_{\Gamma} (\sigma \omega)^T n d\Gamma \]

where

\[ \nabla^T = \begin{pmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{pmatrix} \]

and the tensor \( \sigma \) is in vector form on the left hand side, i.e.,

\[ \sigma^T = ( \sigma_{xx} \ \sigma_{yy} \ \sigma_{zz} \ \sigma_{xy} \ \sigma_{yz} \ \sigma_{xz} ) \]

and in matrix form on the right-hand side, i.e.,

\[ \sigma = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{pmatrix} \]

Proof:

Without loss of generality omit \( \omega \) (otherwise define \( \tilde{\sigma} = \sigma \omega \)). Using the divergence theorem (Green-Gauss theorem) [Heu81, p.522]

\[ \int_{\Omega} \nabla^T \mathbf{F} d\Omega = \int_{\Gamma} \mathbf{F}^T n d\Gamma \]

which in component form in 3D is

\[ \int_{\Omega} \left( \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} \right) d\Omega = \int_{\Gamma} (F_x n_x + F_y n_y + F_z n_z) d\Gamma \]

gives

\[ \int_{\Omega} \nabla^T \sigma d\Omega = \int_{\Omega} \begin{pmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{pmatrix} \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{pmatrix} d\Omega \]

\[ = \int_{\Omega} \left( \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} \right) d\Omega \]

\[ = \int_{\Gamma} \left( \sigma_{xx} n_x + \sigma_{xy} n_y + \sigma_{xz} n_z \right) d\Gamma \]

\[ + \int_{\Gamma} \left( \sigma_{xy} n_x + \sigma_{yy} n_y + \sigma_{yz} n_z \right) d\Gamma \]

\[ + \int_{\Gamma} \left( \sigma_{xz} n_x + \sigma_{yz} n_y + \sigma_{zz} n_z \right) d\Gamma \]
\[
\begin{align*}
  &\quad = \int_{\Gamma} \begin{pmatrix}
\sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\
\sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\
\sigma_{xz} & \sigma_{yz} & \sigma_{zz}
\end{pmatrix}
  \begin{pmatrix}
n_x \\
n_y \\
n_z
\end{pmatrix}
d\Gamma \\
  &\quad = \int_{\Gamma} \mathbf{\sigma}^T \mathbf{n} d\Gamma
\end{align*}
\]

This completes the proof.
A-V atrioventricular
AMLIC anisotropy modulated line integral convolution
AS aortic stenosis
AVS Application Visualization System [AVS]
BPI blood pool imaging
CAD computer aided design
CAM computer aided manufacturing
CCD charge-coupled devices
CFD computational fluid dynamics
CHF congestive heart failure
CIE Commission International de l’Eclairage [=International Commission on Illumination] (also used as a name of a common colour space)
CSF cerebral spinal fluid
CSPAMM complementary spatial modulation of magnetization (tagged MRI pulse sequence) [SSF+99]
CT computed tomography
DANTE delays alternating with nutations for tailored excitation (MRI pulse sequence) [MCD+96]
DDA digital differential analysis
**Abbreviations**

**DENSE** fast displacement encoding with stimulated echoes (MRI pulse sequence)  
[ABW99]

**DNA** deoxyribonucleic acid

**DTI** diffusion tensor imaging

**DVR** direct volume rendering

**DWI** diffusion-weighted (MRI) imaging

**EBCT** electron beam computed tomography [POCD99]

**ECG** electrocardiogram/echocardiography

**ED** end-diastole

**EEG** electroencephalograph/electroencephalography

**EF** ejection fraction

**EKG** electrocardiogram

**ES** end-systole

**FE** finite element

**FEA** finite element analysis

**FEM** finite element method

**FIESTA** fast imaging employing steady-state acquisition (MRI pulse sequence)  
[LSM+02]

**FLASH** fast low flip-angle shot (MRI pulse sequence) [MCD+96]

**FLIC** fast line integral convolution [SH95]

**FLTK** Fast Light Toolkit [Spi]

**fMRI** functional magnetic resonance imaging

**FROLIC** fast rendering of oriented line integral convolution [WG97]

**GLU** OpenGL utility library

**GLUT** OpenGL utility toolkit

**GUI** graphical user interface

**HARP** harmonic phase (MRI pulse sequence) [OKMP99]

**HLS** hue-lightness-saturation (colour space)
HSB  hue-saturation-brightness (colour space)
HSV  hue-saturation-value (colour space)
IDL  Interactive Data Language [Res]
LA  long-axis
LV  left ventricle/left ventricular
LAD  left anterior descending coronary artery
LIC  line integral convolution [CL93]
LVH  left ventricular hypertrophy
MI  myocardial infarction
MEG  magnetoencephalograph/magnetoencephalography
MIP  maximum intensity projection
MPI  myocardial perfusion imaging
MR  magnetic resonance
MRA  magnetic resonance angiography
MRI  magnetic resonance imaging
MRS  magnetic resonance spectroscopy [TM01]
NURBS  non-uniform rational B-Spline
OCT  optical coherence tomography
ODE  ordinary differential equation
OLIC  oriented line integral convolution [WG97]
PC-MRI  phase-contrast MRI
PACS  picture archiving and communication system
PCA  phase-contrast angiography
PD  Parkinson disease
PET  positron emission tomography
PLIC  pseudo line integral convolution [VKP99]
PVL  periventricular leukomalacia
Abbreviations

RGB  red-green-blue (colour space)
RNA  ribonucleic acid
RV   right ventricle/right ventricular
SA   short-axis
S-A  sino-atrial
SD   standard deviation
SPAMM spatial modulation of magnetization (tagged MRI pulse sequence) [YICA94]
SPECT single-photon emission computed tomography
SRI  strain rate imaging
SV   stroke volume
SVC  superior vena cava
SVCS superior vena cava syndrome
SVD  singular value decomposition
TEE  transesophageal echocardiography
UFLIC unsteady flow line integral convolution [SK97b, SK98]
UML  unified modeling language
US   ultrasonography
VEC-MRI velocity-encoded cine MRI
VR   Virtual Reality
VRML Virtual Reality Modelling Language
VTK  Visualization Toolkit [Kit]
XML  extended markup language
This appendix contains an explanation of technical terms used in this thesis which might not be known to the reader. Since we assume that the reader is not a medical specialist this glossary contains mostly medical terms though some engineering and mathematical terms are listed where deemed appropriate. Good medical online dictionaries are [Hey95, Har, Web02].

±2 standard deviation a deviation of $2\sigma$ from either side of the mean of a standard normal distribution. The area under the Gaussian curve within these boundaries is

$$\text{area} = \int_{-2}^{2} \frac{e^{-x^2/2}}{\sqrt{2\pi}} dx = 0.954$$

i.e., 95.4% of all data values for a standard normal distribution lie within $2\sigma$ of the mean [Lar82].

**A-V node** see *atrioventricular node*

**aberrant** an atypical group, individual, or structure, especially one with an aberrant chromosome number [McG].

**active contour model** (also *snake*) is an energy-minimizing spline guided by external constraint forces that pull it towards features in a set of images such as lines and edges.

**acoustic impedance** absorption of sound in a medium, equal to the ratio of the sound pressure at a boundary surface to the sound flux through the surface [Enc].

**akinesia** loss or impairment of voluntary activity (as of a muscle) [Web02].

**akinetic** of, relating to, or affected by *akinesia* [Web02].
anatomy 1: a branch of morphology that deals with the structure of organisms (comp. physiology).
2: the art of separating the parts of an organism in order to ascertain their position, relations, structure, and function.
3: structural makeup especially of an organism or any of its parts [Web02].

aneurysm (Gr. aneurysma a widening) a sac formed by the dilation of the wall of an artery, a vein, or the heart. The chief signs of arterial aneurysm are the formation of a pulsating tumour, and often a bruit (aneurysmal bruit) heard over the swelling. Sometimes there are symptoms from pressure on contiguous parts [Hey95].

angiogenesis formation and differentiation of blood vessels [Web02].

angiography 1: the roentgenographic visualization of the blood vessels after injection of a radiopaque substance [Web02].
2: phase-contrast angiography: see phase-contrast MRI.

angiocardiography an X-ray examination of the heart and great vessels following the injection of a radiopaque contrast medium into a blood vessel or one of the cardiac chambers [Har].

anterior towards the abdominal surface of the body (Brain anatomy: towards the face) [EW91].

aorta the large arterial trunk that carries blood from the heart to be distributed by branch arteries through the body [Web02].

apex a narrowed or pointed end of an anatomical structure: as (a) the narrow somewhat conical upper part of a lung extending into the root; (b) the lower pointed end of the heart situated in humans opposite the space between the cartilages of the fifth and sixth ribs on the left side; (c) the extremity of the root of a tooth [Web02].

apical of, relating to, or situated at an apex [Web02].

artery any of the tubular branching muscular- and elastic-walled vessels that carry blood from the heart through the body [Web02].

atheroma (adj. atheromatous) 1: fatty degeneration of the inner coat of the arteries.
2: an abnormal fatty deposit in an artery [Web02].

atherosclerosis narrowing of an artery; characterized by atheromatous deposits in and fibrosis of the inner layer of arteries [Web02].

atrioventricular node a small mass of tissue that is situated in the wall of the right atrium adjacent to the septum between the atria, passes impulses received
from the sinoatrial node to the ventricles by way of the bundles of His, and in some pathological states replaces the sinoatrial node as pacemaker of the heart [Web02].

**auscultation** the act of listening to sounds arising within organs (as the lungs or heart) as an aid to diagnosis and treatment [Web02].

**autonomic** 1: (a) acting or occurring involuntary <autonomic reflexes> (b) relating to, affecting, or controlled by the autonomic nervous system <autonomic dysfunction>.  
2: having an effect upon tissue supplied by the autonomic nervous system <autonomic drugs> [Web02].

**autonomic nervous system** (also vegetative nervous system) a part of the vertebrate nervous system that innervates smooth and cardiac muscle and glandular tissues and governs involuntary actions (as secretion, vasoconstriction, or peristalsis) and that consists of the sympathetic nervous system and the parasympathetic nervous system (compare central nervous system, peripheral nervous system) [Web02].

**axon** a usually long and single nerve-cell process that usually conducts impulses away from the cell body [Web02].

**basal** 1: relating to, situated at, or forming the base.  
2: of, or relating, or essential for maintaining the fundamental vital activities of an organism (as respiration, heartbeat, or excretion) [Web02].

**basal ganglion** (also basal nucleus) any of four deeply placed masses of gray matter within each cerebral hemisphere comprising the caudate nucleus, the lentiform nucleus, the amygdala, and the claustrum - usually used in plural [Web02].

**cardiac** (L. cardiacus from Gr. kardiakos) pertaining to the heart [Hey95].

**cardiac blood pool imaging** this noninvasive test uses radioactive tracers to delineate the heart’s chambers and major vessels. It may be used to detect a heart attack, heart muscle function, and coronary artery disease. The patient receives a radioactive tracer by injection (into a vein) and then the heart is imaged using a gamma camera. The heart is imaged before and after exercise. This test may be used to detect and evaluate atrial septal defect, congestive heart failure, cardiomyopathy, Lyme disease (secondary), mitral stenosis, and superior vena cava syndrome [Meda].

**cardiac output** the volume of blood ejected from the left side of the heart in one minute - called also minute volume [Web02].

**cardiomyopathy** disease of the heart muscle that impairs the ability of the heart to pump [Maya].
cardiovascular pertaining to the heart and blood vessels [Hey95].

caudate nucleus (also caudate) the one of the four basal ganglia in each cerebral hemisphere that comprises a mass of gray matter in the corpus striatum, forms part of the floor of the lateral ventricle, and is separated from the lentiform nucleus by the internal capsule [Web02].

central nervous system the part of the nervous system which in vertebrates consists of the brain and spinal cord, to which sensory impulses are transmitted and from which motor impulses pass out, and which supervises and coordinates the activity of the entire nervous system (compare autonomic nervous system, peripheral nervous system) [Web02].

cerebellum portion of the brain responsible for coordinating movements [Maya].

cerebral cortex the superficial layer of the cerebral hemispheres, composed of gray matter and concerned with coordination of higher nervous activity [McG].

cerebrum largest portion of the brain, consisting of two hemispheres; responsible for thinking, feelings and voluntary movement [Maya].

cholangiopancreatography technique for imaging the pancreatobiliary tree [Mat].

chorda tendinea any of the delicate tendinous cords that are attached to the edges of the atroioventricular valves of the heart and to the papillary muscles and serve to prevent the valves from being pushed into the atrium during the ventricular contraction [Web02].

ciliary ganglion a ganglion which lies directly behind the eye [Unic].

cine a series of rapidly recorded multiple images taken at sequential cycles of time and displayed on a monitor in a dynamic movie display format [Fon].

cognition 1: cognitive mental processes.
2: a conscious intellectual act [Web02].

cognitive of, relating to, or being conscious intellectual activity (as thinking, reasoning, remembering, imagining, or learning words) [Web02].

collagen (Gr. kolla glue + gennan to produce) the protein substance of the white fibres (collagenous fibres) of skin, tendon, bone, cartilage, and all other connective tissue; composed of molecules of tropocollagen (q.v.), it is converted into gelatin by boiling [Hey95].

collagenous pertaining to collagen; forming or producing collagen [Hey95].

collateral vessel can be pre-existing vessels that normally have little or no blood flow. Acute occlusion of normal vessels (e.g., thrombosis of a large artery) can cause a redistribution of pressures within the vascular bed thereby causing
blood flow to occur in collateral vessels. Conditions of chronic stress (e.g., endur-
ance exercise training) can cause new blood vessels to form by angiogenesis [Kla].

collateralization structural development of existing vessels which provide alterna-
tive blood supply routes after occlusion of another artery [Unt03].

computed tomography (CT) also called CT or CAT scan: X-ray technique that
uses a computer to construct images of the body [Maya].

congenital (L. congenitus born together) existing at, and usually before, birth;
referring to conditions that are present at birth, regardless of their causation
[Hey95].

congestion (L. congestio, from congerere to heap together) excessive or abnormal
accumulation of blood in a part [Hey95].

congestive heart failure (CHF) occurs when muscle cells in the heart die or no
longer function properly, causing the heart to lose its ability to pump enough
blood through the body. Heart failure usually develops gradually, over many
years, as the heart becomes less and less efficient. Systolic heart failure occurs
when the heart’s ability to contract decreases. The heart cannot pump with
enough force to push a sufficient amount of blood through the body. Blood
coming into the heart from the lungs may back up and cause fluid to leak into
the lungs, a condition known as pulmonary congestion. Diastolic heart failure
occurs when the heart has difficulty relaxing (it’s too efficient) and cannot
properly fill with blood because the muscle has become stiff. This may lead
to fluid accumulation, especially in the feet, ankles, legs and lungs [Medb].

constrictive pericarditis when the pericardium is scarred or thickened, the heart
has difficulty contracting. This is because the pericardium has shrunken or
tightened around the heart, constricting the muscle’s heart movement. This
usually occurs as a result of tuberculosis, which now is rarely found in the
United States, except in immigrant, AIDS, and prison populations [Medb].

corticospinal tract any of four columns of motor fibers of which two run on each
side of the spinal cord and which are continuations of the pyramids of the
medulla oblongata: (a) lateral corticospinal tract (b) ventral corticospinal tract
[Web02].

cranium portion of the skull that houses the brain [Maya].

cytoplasm the organized complex of inorganic and organic substances external to
the nuclear membrane of a cell and including the cytosol and membrane-bound
organelles (as mitochondria or chloroplasts) [Web02].

dendrite (Gr. dendron tree) the highly branched tree-like process of a neuron that
serves as a receptive field and conducts impulses toward the cell body [Bru].
deoxyribonucleic acid (DNA) the molecule that encodes genetic information. DNA is a double-stranded molecule held together by weak bonds between base pairs of nucleotides. The four nucleotides in DNA contain the bases: adenine (A), guanine (G), cytosine (C), and thymine (T). In nature, base pairs form only between A and T and between G and C; thus the base sequence of each single strand can be deduced from that of its partner [Sch].

dependent variable variable defined over the data domain [WLG97].

diabetes any of various abnormal conditions characterized by the secretion and excretion of excessive amounts of urine; especially: diabetes mellitus [Web02].

diabetes mellitus a variable disorder of carbohydrate metabolism caused by a combination of hereditary and environmental factors and usually characterized by inadequate secretion or utilization of insulin, by excessive urine production, by excessive amounts of sugar in the blood and urine, and by thirst, hunger, and loss of weight [Web02].

diastole Period during the heart cycle in which the muscle relaxes, followed by contraction (systole). At end-diastole the heart has achieved maximum filling [Maya, GZM97].

diffusion 1: the process whereby particles of liquids, gases, or solids intermingle as the result of their spontaneous movement caused by thermal agitation and in dissolved substances move from a region of higher to one of lower concentration. 2 (a) reflection of light by a rough reflecting surface (b) transmission of light through a translucent material [Web02].

dilated cardiomyopathy a condition in which the heart’s ability to pump blood is reduced because the left ventricle (one of the two pumping chambers of the heart) is enlarged (dilated) [Hea].

Doppler effect a change in the frequency with which waves (as sound, light, or radio waves) from a given source reach an observer when the source and the observer are in motion with respect to each other so that the frequency increases or decreases according to the speed at which the distance is decreasing or increasing [Web02].

dysplasia (dys- + Gr. plassein to form) abnormality of development; in pathology, alteration in size, shape, and organization of adult cells [Hey95].

echocardiography ultrasonography applied to the heart.

edema an abnormal excess accumulation of serous fluid in connective tissue or in a serous cavity [Web02].

ejection fraction a measure of ventricular contractility, equal to normally 65.8%; lower values indicate ventricular dysfunction [Hey95].
**electrocardiogram (ECG or EKG)** a graphical record of the variations in electrical potential caused by electrical activity of the heart muscle and detected at the body surface, as a method for studying the action of the heart muscle [Hey95].

**electrocardiography** the making of an electrocardiogram [Hey95].

**electroencephalogram** a graphic record of minute changes in the electric potential associated with the activity of the cerebral cortex, as picked up by electrodes placed on the scalp [Har].

**electroencephalography** a method of graphically recording brain wave activity, used to diagnose seizure disorders, brainstem disorders, tumors, or clots [Har].

**embolus** an abnormal particle (as an air bubble) circulating in the blood [Web02].

**endocrine** (endo- + Gr. krinein to separate) pertaining to internal secretions; hormonal [Hey95].

**endothelium** cell layer forming the interface between the blood and vessel walls. The endothelium plays a critical role in the mechanics of blood flow, vascular smooth muscle cell growth, and serves as a barrier to the transvascular diffusion of liquids and solutes [URL: http://hsc.virginia.edu/medicine/basic-sci/biomed/ley/endothelium.htm].

**endocarditis** exudative and proliferative inflammatory alterations of the endocardium, characterized by the presence of vegetations on the surface of the endocardium or in the endocardium itself, and most commonly involving a heart valve, but sometimes affecting the inner lining of the cardiac chambers or the endocardium elsewhere. It may occur as a primary disorder or as a complication of or in association with another disease [Hey95].

**endocardium** the thin, inner membrane that lines the heart [Maya].

**enzyme** any of numerous complex proteins that are produced by living cells and catalyze specific biochemical reactions at body temperatures [Web02].

**epicardium** the thin membrane on the surface of the heart [Maya].

**esophagus** a muscular tube that in adult humans is about nine inches (23cm) long and passes from the pharynx down the neck between the trachea and the spinal column and behind the left bronchus where it pierces the diaphragm slightly to the left of the middle line and joins the cardiac end of the stomach [Web02].

**excitation** delivering (inducing, transferring) energy into a “spinning” nuclei via radio-frequency pulse(s), which puts the nuclei into a higher energy state. By producing a net transverse magnetization an MRI system can observe a response from the excited system [Fon].
fasciculus (diminutive of Lat. fascic = bundle) bundle of nerve or muscle fibers [AB].

fibrillation a small, local, involuntary contraction of muscle, invisible under the skin, resulting from spontaneous activation of single muscle cells or muscle fibres [Hey95].

fibrosis the formation of fibrous tissue; fibroid or fibrous degeneration [Hey95].

fibrous 1: containing, consisting of, or resembling fibers <collagen is a fibrous protein>.
2: characterized by fibrosis [Web02].

fluoroscopy observing the internal structure of opaque objects (as the living body) by means of the shadow cast by the object examined upon a fluorescent screen when placed between the screen and a source of X rays [Web02].

Fragile X syndrome (also Martin-Bell syndrome, Marker X syndrome, FRAXA syndrome) is the most common form of inherited mental retardation. Individuals with this condition have developmental delay, variable levels of mental retardation, and behavioral and emotional difficulties. They may also have characteristic physical traits. Generally, males are affected with moderate mental retardation and females with mild mental retardation. Fragile X syndrome is caused by a mutation in the FMR-1 gene, located on the X chromosome [Medb].

functional imaging can identify the kinds of molecular structures/ receptors that cover the surface of e.g., a tumor, information that potentially can predict how it may behave and respond to certain treatments. By providing a picture of glucose utilization in tumor cells, imaging can demonstrate without the need for a biopsy how a tumor is responding to a recently administered treatment [CHIa].

ganglion a collection of nerve cell bodies outside the central nervous system. They can be associated with the autonomic nervous system, with cranial nerves, or spinal nerves [Unic].

Gaussian curvature an intrinsic property of space independent of the coordinate system used to describe it. The Gaussian curvature of a regular surface in $\mathbb{R}^3$ is

$$K = \kappa_1 \kappa_2$$

where $\kappa_1$ and $\kappa_2$ are the principal curvatures of the surface [Wei].

gene expression the process by which a gene’s coded information is converted into the structures present and operating in the cell. Expressed genes include those that are transcribed into mRNA (messenger RNA) and then translated into protein and those that are transcribed into RNA but not translated into protein [Sch].
heat flux the flow of heat (transfer of energy from one substance to another as a result of a temperature difference) across a surface of unit area in a unit amount of time; commonly expressed in units of cal/(cm²sec) or W/m² [Har].

His-Purkinje system (also bundle of His, atrioventricular bundle, His bundle) a slender bundle of modified cardiac muscle that passes from the atrioventricular node in the right atrium to the right and left ventricles by way of the septum and that maintains the normal sequence of the heartbeat by conducting the wave of excitation from the right atrium to the ventricles [Web02].

hypertension (hyper- + tension) persistently high arterial blood pressure. Various criteria for its threshold have been suggested, ranging from 140 mm Hg systolic and 90 mm Hg diastolic to as high as 200 mm Hg systolic and 110 mm Hg diastolic. Hypertension may have no known cause (essential or idiopathic h.) or be associated with other primary diseases (secondary h.) [Hey95].

hypertensive marked by a rise in blood pressure: suffering or caused by hypertension [Web02].

hypertrophic relating to or affected by hypertrophy [Har].

hypertrophy (hyper- + Gr. troph nutrition) the enlargement or overgrowth of an organ or part due to an increase in size of its constituent cells [Hey95].

hypokineti c characterized by, associated with, or caused by decreased motor activity [Web02].

hypothalamus a basal part of the forebrain that lies beneath the thalamus on each side, forms the floor of the third ventricle, and includes vital autonomic regulatory centers (as for the control of food intake) [Web02].

idiopathic arising spontaneously or from an obscure or unknown cause [Cru].

independent variable 1: a variable that is deliberately varied or changed in a controlled manner in an experiment to observe its effects on the response variable; sometimes caused causal variable [ERC].
2: a variable representing the data domain [WLG97].

infarct an area of tissue that dies because of lack of blood supply [Maya].

infarction 1: the process of forming an infarct.
2: see infarct [Web02].

inferior towards the feet (Brain anatomy: towards the base of the skull) [EW91].

inotropic relating to or influencing the force of muscular contractions <digitalis is a positive inotropic agent> [Web02].
**intima** the innermost coat of an organ (as a blood vessel) consisting usually of an endothelial layer backed by connective tissue and elastic tissue - called also *tunica intima* [Web02].

**in vitro** within a glass; observable in a test tube; in an artificial environment [Hey95].

**in vivo** within the living body [Hey95].

**ischemia** deficiency of blood flow within an organ or part of an organ [Maya].

**ischemic** oxygen-starved [GZM97].

**isoparametric element** finite element for which the basis functions and the geometry functions coincide.

**kinetic** (Gr. *kintikos*) pertaining to or producing motion [Hey95].

**lateral** of or relating to the side; especially of a body part: lying at or extending toward the right or left side: lying away from the median axis of the body [Web02].

**lateral corticospinal tract** (also *crossed pyramidal tract*) a band of nerve fibers that descends in the posterolateral part of each side of the spinal cord and consists mostly of fibers arising in the motor cortex of the contralateral side of the brain and crossing over in the decussation of pyramids with some fibers arising in the motor cortex of the same side [Web02].

**ligament** 1: a tough band of tissue that serves to connect the articular extremeties of bones or to support or retain an organ in place and is usually composed of coarse bundles of dense white fibrous tissue parallel or closely interlaced, pliant, and flexible, but not extensible.
   2: any of various folds or bands of membranes connecting parts or organs [Web02].

**Lyme disease** (also called *Lyme, Lyme borreliosis*) an acute inflammatory disease that is usually characterized initially by the skin lesion erythema migrans and by fatigue, fever, and chills and if left untreated may later manifest itself in cardiac and neurological disorders, joint pain, and arthritis and that is caused by a spirochete of the genus *Borrelia* (*B. burgdorferi*) transmitted by the bite of a tick especially of the genus *Ixodes* (*I. scapularis* syn. *I. dammini* in the eastern and midwestern U.S., *I. pacificus* especially in some parts of the Pacific coastal states of the U.S., and *I. ricinus* in Europe) [Web02].

**magnetic haemodynamic effect** see *magnetohydrodynamic effect.*
magnetic resonance imaging (MRI) a noninvasive diagnostic technique that produces computerized images of internal body tissues and is based on nuclear magnetic resonance of atoms within the body induced by the application of radio waves [Web02].

magnetic resonance spectroscopy (MRS) provides a measure of metabolic differences in various brain areas, as in detecting foci of acute cerebral ischemia and stroke [TM01].

magnetohydrodynamic effect (also magnetic haemodynamic effect) generation of an electric current when a conducting fluid (blood) moves in a magnetic field. It causes an elevation of the T wave of a patient’s ECG during MRI [SB99, WK98].

magnetoencephalograph an instrument that records magnetic signals proportional to electroencephalographic waves emanating from electrical activity in the brain [Har].

magnetoencephalography the making of an magnetoencephalograph [Har].

mean curvature defined as

$$H = \frac{1}{2} (\kappa_1 + \kappa_2)$$

where $K = \kappa_1$ and $\kappa_2$ are the principal curvatures [Wei].

metabolic of, relating to, or based on metabolism [Web02].

metabolism the sum of the processes in the buildup and destruction of protoplasm; specifically: the chemical changes in living cells by which energy is provided for vital processes and activities and new material is assimilated [Web02].

metastasis change of position, state, or form: as
1: a secondary metastatic growth of a malignant tumor.
2: transfer of a disease-producing agency (as cancer cells or bacteria) from an original site of disease to another part of the body with development of a similar lesion in the new location [Web02].

mitral stenosis a condition usually the result of disease in which the mitral valve is abnormally low [Web02].

molecular imaging involves the use of molecular probes or tracers. Molecular Imaging fuses the disciplines of molecular biology, genetic engineering, immunology, cytology, and biochemistry with imaging. Advances in MRI/MRS, MR microscopy, cellular tags, PET and SPECT are used to evaluate normal and abnormal tissue metabolism and perfusion in response to genetic, physiological, or therapeutic challenges [CHIa].
**multiple sclerosis** a demyelinating disease marked by patches of hardened tissue in the brain or the spinal cord and associated especially with partial or complete paralysis and jerking muscle tremor [Web02].

**myelin** (also *white substance of Schwann*) the substance of the cell membrane of Schwann’s cells that coils to form the myelin sheath [Har].

**myelin sheath** a layer of fatty material (*myelin*) that surrounds certain nerve fibers; it has a high proportion of lipid to protein, and serves as an electrical insulator [Har].

**myocarditis** (*myo- + Gr. *kardia* heart + -itis*) inflammation of the myocardium; inflammation of the muscular walls of the heart [Hey95].

**myocardial infarction** heart attack; death of an area of heart muscle due to lack of blood supply [Maya].

**myocardial perfusion imaging** a nuclear medicine technique designed to depict the blood flow to the heart muscle noninvasively. The radioactive drugs (radiopharmaceuticals) used for this study are extracted by the heart muscle in proportion to the local blood flow in the heart [UNM01].

**myocardium** the heart muscle [Maya].

**myocyte** 1: any contractile cell. 2: a muscle cell [Har].

**myopathy** (*myo- + -pathy*) any disease of a muscle [Hey95].

**myxoma** (also *myxoblastoma*) a soft, jellylike tumor that is composed of connective and mucoid tissue and that may grow to over 30 cm in diameter [Har].

**near-infrared fluorescence imaging** imaging modality which uses a probe emitting a strong near-infrared signal after activation (e.g., by tumour proteases) which is registered by a photodetector (e.g., CCD camera) [WTMB99].

**necrosis** (*Gr. *nekrosis* deadness) the sum of the morphological changes indicative of cell death and caused by the progressive degradative action of enzymes; it may affect groups of cells or part of a structure or an organ [Hey95].

**neuro** (also *neurological*) of or relating to the nervous system especially in respect to its structure, functions and abnormalities [Web02].

**neuron** a nerve cell: the functional unit of the nervous system. Structurally, the neuron is made up of a cell body (soma) and one or more long processes: a single axon and dendrites [McG].
neuroglia  supporting tissue that is intermingled with the essential elements of nervous tissue especially in the brain, spinal cord, and ganglia, is of ectodermal origin, and is composed of a network of fine fibrils and of flattened stellate cells with numerous radiating fibrillar processes [Web02].

non-uniform rational B-Spline (NURBS)  NURBS curves and surfaces are the industry standard for geometry description in CAD/CAM or Computer Graphics. NURBS are piecewise rational; they are the logical extension of B-Splines or Bezier entities [Far95].

occipital of, relating to, or located within or near the occiput or the occipital bone [Web02].

optic radiation any of several neural radiations concerned with the visual function; especially: one made up of fibers from the pulvinar and the lateral geniculate body to the cuneus and other parts of the occipital lobe [Web02].

optical coherence tomography (OCT)  is a new imaging technique that utilizes photonics and fiber optics to obtain images and tissue characterization. OCT uses infrared light waves that reflect off the internal microstructure within the biological tissues. The frequencies and bandwidths of infrared light are orders of magnitude higher than medical ultrasound signals resulting in greatly increased image resolution [Lig].

optical imaging the branch of in vivo diagnostics that generates images by using photons of light in the wavelength range from ultraviolet to near-infrared, including the range of wavelengths visible to the human eye. The propagation of light through tissue is mainly influenced by absorption and scattering in the tissue itself or by a fluorescent contrast agent [WWH01].

orthogonal transformation a linear transformation $T : V \to V$ which preserves a symmetric inner product. In particular, an orthogonal transformation (technically, an orthonormal transformation) preserves lengths of vectors and angles between vectors,

$$\langle v, w \rangle = \langle Tv, Tw \rangle$$

In addition, an orthogonal transformation is either a rigid rotation or a rotoinversion (a rotation followed by a flip). Orthogonal transformations correspond to and may be represented using orthogonal matrices [Wei].

palpation examination by touching [Maya].

pancreatobiliary tree consists of the bile ducts, which are tubes that carry bile from the liver to the gallbladder and small intestine, and the pancreas which is a large gland that produces chemicals that help with digestion [URL: http://ww w.niddk.nih.gov/health/digest/pubs/diagtest/ercp.htm].
papillary muscle one of the small muscular columns attached at one end to the chordae tendineae and at the other to the wall of the ventricle and that maintain tension on the chordae tendineae as the ventricle contracts [Web02].

paracellular surrounding the cells [DBK01].

parasympathetic nervous system the part of the autonomic nervous system that contains chiefly cholinergic fibers, that tends to induce secretion, to increase the tone and contractility of smooth muscle, and to slow the heart rate, and that consists of (1) a cranial part made up of preganglionic fibers leaving and passing the midbrain by the oculomotor nerves and the hindbrain by the facial, glossopharyngeal, vagus, and accessory nerves and passing to the ciliary, sphenopalatine, submandibular, and optic ganglia of the head or to ganglionated plexuses of the thorax and abdomen and postganglionic fibers passing from these ganglia to end organs of the head and upper trunk and (2) a sacral part made up of preganglionic fibers emerging and passing in the sacral nerves and passing to ganglionated plexuses of the lower trunk and postganglionic fibers passing from these plexuses chiefly to the viscera of the lower abdomen and the external genital organs [Web02].

parietal 1: of or relating to the walls of a part or cavity.
2: of, relating to, or located in the upper posterior part of the head; specifically: relating to the parietal bones [Web02].

Parkinson disease (PD) is a progressive movement disorder marked by tremors, rigidity, slow movements (bradykinesia), and posture instability. It occurs when cells in the substantia nigra, one of the movement-control centers of the brain, begin to die for unknown reasons. Most cases of PD are sporadic. This means that there is a spontaneous and permanent change in nucleotide sequences (the building blocks of genes). PD was first noted by British physician James Parkinson in the early 1800s [Medb].

pathology branch of medicine that treats the essential nature of the disease, especially the structural and functional changes in tissues and organs of the body caused by the disease [Hey95].

pathophysiologic 1: the physiology of abnormal states; specifically
2: the functional changes that accompany a particular syndrome or disease [Web02].

perfusion 1: the act of pouring over or through, especially the passage of a fluid through the vessels of a specific organ.
2: a liquid poured over or through an organ or tissue [Hey95].

perception awareness of the elements of environment through physical sensation [Web02].
pericardium  the conical sac of serous membrane that encloses the heart and the roots of the great blood vessels of vertebrates and consists of an outer fibrous coat that loosely invests the heart and is prolonged on the outer surface of the great vessels except the inferior vena cava and a double inner serous coat of which one layer is closely adherent to the heart while the other lines the inner surface of the outer coat with the intervening space being filled with pericardial fluid [Web02].

pericarditis  is an inflammation of the two layers of the pericardium, the thin, sac-like membrane that surrounds the heart [Medb].

peripheral nervous system  the part of the nervous system that is outside the central nervous system and comprises the cranial nerves excepting the optic nerve, the spinal nerves, and the autonomic nervous system [Web02].

periventricular leukomalacia (PVL)  damage or softening of the white matter near the ventricles [Chib].

phase-contrast MRI (PC-MRI)  a magnetic resonance imaging (MRI) scan performed after the injection of a contrast medium (a dye that makes blood vessels more visible). PC-MRI may be better than an angiogram at detecting whether a vessel has closed after an angioplasty [Hea].

physiology  1: a branch of biology that deals with the functions and activities of life or of living matter (as organs, tissues, or cells) and of the physical and chemical phenomena involved.
2: the organic processes and phenomena of an organism or any of its parts or of a particular bodily process (e.g., the physiology of the thyroid gland) [Web02].

polarization  [particle physics] property of a collection of particles with spin, in which the majority have spin components pointing in one direction, rather than at random [McG].

positron emission tomography (PET)  tomography in which an in vivo, noninvasive, cross-sectional image of regional metabolism is obtained by a usually color-coded cathode-ray tube representation of the distribution of gamma radiation given off in the collision of electrons in cells with positrons emitted by radionuclides incorporated into metabolic substances

posterior  towards the back of the body (brain anatomy: towards the occipital region) [EW91].

principal curvature  the maximum and minimum of the normal curvature $\kappa_1$ and $\kappa_2$ at a given point on a surface are called the principal curvatures. The normal curvature $\kappa_n$ is the curvature of the curve created by cutting the surface at the given point with a plane which contains the normal at that point. Formulas for the computation of the principal curvatures are given in [Wei, HL92b].
prolate spheroidal coordinates a system of curvilinear coordinates in which two sets of coordinate surfaces are obtained by revolving the curves of elliptic cylindrical coordinates about the x-axis, which is relabeled the z-axis. The third set of coordinate planes consists of planes passing through this axis:

\[
\begin{align*}
  x &= a \sinh \xi \sin \nu \cos \theta \\
  y &= a \sinh \xi \sin \nu \sin \theta \\
  z &= a \cosh \xi \cos \nu
\end{align*}
\]

where \( \xi \in [0, \infty) \), \( \nu \in [0, \pi] \), and \( \theta \in [0, 2\pi) \) [Wei].

protease any of numerous enzymes that hydrolyze proteins and are classified according to the most prominent functional group (as serine or cysteine) at the active site - called also proteinase [Web02].

psychology 1: the science of mind and behaviour.
2: the mental or behavioral characteristics typical of an individual or group or a particular form of behavior; the study of mind and behavior in relation to a particular field of knowledge or activity [Web02].

pulmonary artery an arterial trunk or either of its two main branches that carry blood to the lungs: (a) a large arterial trunk that arises from the conus arteriosus of the right ventricle, ascends in front of the aorta, and branches into the right and left pulmonary arteries – called also pulmonary trunk (b) a branch of the pulmonary trunk that passes under the arch of the aorta to the right lung where it divides into branches – called also right pulmonary artery (c) a branch of the pulmonary trunk that passes to the left in front of the descending part of the aorta, gives off the ductus arteriosus in the fetus which regresses to the ligamentum arteriosum in the adult, and passes to the left lung where it divides into branches - called also left pulmonary artery [Web02].
pulmonary vein any of usually four veins comprising two from each lung that return oxygenated blood from the lungs to the superior part of the left atrium, that may include three veins from the right lung if the veins from all three lobes of the right lung remain separate, and that may include a single trunk from the left lung if its major veins unite before emptying into the left atrium [Web02].

pyramidal tract see corticospinal tract

radiography (radio- + Gr. graphein to write) the making of film records (radio-graphs) of internal structures of the body by passage of x-rays or gamma rays through the body to act on specially sensitized film [Hey95].

radiopaque being opaque to radiation and especially X-rays [Web02].

regurgitation (re- + L. gurgitare to flood) a backward flowing, as the casting up of undigested food, or the backward flowing of blood into the heart, or between the chambers of the heart when a valve is incompetent [Hey95].

remodelling an enlargement and thinning out of the heart’s left ventricle as it tries to adapt to the effects of heart failure. This leads to further damage to heart cells, reduced cardiac output of blood and more severe heart disease [Hea].

reperfusion restoration of the flow of blood to a previously ischemic tissue or organ (as the heart) [Web02].

revascularization, transmyocardial technique, in which microscopic holes are made into the heart muscle using a laser. It is speculated that such holes improve blood flow to the heart by promoting angiogenesis [Mayb].

ribonucleic acid (RNA) a chemical found in the nucleus and cytoplasm of cells; it plays an important role in protein synthesis and other chemical activities of the cell. The structure of RNA is similar to DNA. There are several classes of RNA molecules, including messenger RNA, transfer RNA, ribosomal RNA, and other small RNAs, each serving a different purpose [Sch].

S-A node see sino-atrial node

sagittal 1: of, relating to, or being the sagittal suture of the skull.
2: of, relating to, situated in, or being the median plane of the body or any plane parallel to it [Web02].

sagittal suture the deeply serrated articulation between the two parietal bones in the median plane of the top of the head [Web02].

saltatory conduction electrical conduction by jumping action potentials. Occurs in myelinated axons which have Schwann cells wrapped so tightly around the axonal membrane that no extracellular space is underneath them. Therefore,
the only place that an action potential can occur is at the node of Ranvier—the space between the Schwann cells [http://lls.stcc.mass.edu/tamarkin/AP/AP1 pages/saltator.htm].

**sarcomere** the basic contractile unit of muscle (skeletal and cardiac) that structurally is the portion of a myofibril between two adjacent Z lines, the unit repeated along the entire length of the myofibril [Har].

**schizophrenia** a psychotic disorder characterized by loss of contact with the environment, by noticeable deterioration in the level of functioning in everyday life, and by disintegration of personality expressed as disorder of feeling, thought, and conduct [Web02].

**scintigraphy** producing a graphic record of the gamma rays emitted by a radioiso- tope, revealing its relative concentration in various tissues [Har].

**septal** of or relating to a **septum** [Web02].

**septal defect** hole in the wall separating the two atria or the two ventricles [Maya].

**septum** a wall dividing two cavities or compartments [Maya].

**serous** of, relating to, producing, or resembling **serum**; especially: having a thin watery constitution <a **serous** exudate> [Web02].

**serum** the watery portion of an animal fluid remaining after coagulation: (a) the clear yellowish fluid that remains from blood plasma after fibrinogen, prothrombin, and other clotting factors have been removed by clot formation - called also blood serum (b) a normal or pathological serous fluid (as in a blister) [Web02]

**single photon emission computed tomography (SPECT)** a medical imaging technique that is used especially for mapping brain function and that is similar to **positron-emission tomography** in using the photons emitted by the agency of a radioactive tracer to create an image but that differs in being able to detect only a single photon for each nuclear disintegration and in generating a lower-quality image [Web02].

**sinus node** see **sino-atrial node**

**sino-atrial node** a small mass of tissue that is made up of Purkinje fibers, ganglion cells, and nerve fibers, that is embedded in the musculature of the right atrium of higher vertebrates, and that originates the impulses stimulating the heartbeat [Web02].

**sonomicrometry** measurement of distances using ultra-sound and transducers made from piezo-electric ceramic material to transmit and receive sound energy. Offers a resolution of up to 0.015mm [http://www.sonometrics.com/sono _101.htm].
**stenosis** the narrowing or closure of an opening or passageway in the body (aortic stenosis = narrowing of the valve opening between the left ventricle and the aorta; mitral stenosis = narrowing of the valve between the left atrium and ventricle) [Maya].

**sternal** of or relating to the sternum [Web02].

**sternum** the breastbone [Maya].

**stethoscope** an instrument that is used in medical observation to transmit low-volume internal bodily sounds, such as the heartbeat, or intestinal, venous, or fetal sounds, to the ear of the observer; it consists of two earpieces connected by means of flexible tubing to a diaphragm, which is placed against the patient’s skin [Har].

**strain rate imaging (SRI)** uses ultrasound to measure the rate of deformation of tissue using a method based on the Doppler technique. In cardiac imaging SRI can, unlike tissue Doppler, differentiate between actively deforming and passively drawn muscle segments [Hei].

**stunned myocardium** reversible postischemic mechanical dysfunction that occurs after reperfusion despite the absence of irreversible damage [SSW+96].

**subendocardial** situated or occurring beneath the endocardium or between the endocardium and myocardium [Web02].

**subepicardial** situated or occurring beneath the epicardium or between the epicardium and myocardium [Web02].

**substantia nigra** a layer of deeply pigmented gray matter situated in the midbrain and containing the cell bodies of a tract of dopamine-producing nerve cells whose secretion tends to be deficient in Parkinson’s disease [Web02].

**superior** towards the head (Brain anatomy: towards the top of the head) [EW91].

**superior vena cava** large vein returning blood from the head and arms to the heart [Maya].

**superior vena cava syndrome (SVCS)** (also superior mediastinal syndrome, superior vena cava obstruction) is a partial occlusion of the superior vena cava. This leads to a lower than normal blood flow through this major vein. More than 95% of all cases of SVCS are associated with cancers involving the upper chest. The cancers most commonly associated with SVCS are advanced lung cancers, which account for nearly 80% of all cases of SVCS, and lymphoma [Medb].

**sympathetic nervous system** the part of the autonomic nervous system that is concerned especially with preparing the body to react to situations of stress
or emergency, that contains chiefly adrenergic fibers and tends to depress secretion, decrease the tone and contractility of smooth muscle, increase heart rate, and that consists essentially of preganglionic fibers arising in the thoracic and upper lumbar parts of the spinal cord and passing through delicate white rami communicantes to ganglia located in a pair of sympathetic chains situated one on each side of the spinal column or to more peripheral ganglia or ganglionated plexuses and postganglionic fibers passing typically through gray rami communicantes to spinal nerves with which they are distributed to various end organs [Web02].

**synapses** the place at which a nervous impulse passes from one neuron to another [Web02].

**systemic** pertaining to or affecting the body as a whole [Hey95].

**sytole** the portion of the heart cycle during which the heart muscle is contracting. At end-systole the heart has ejected the maximum amount of blood [Maya, GZM97].

**tachycardia** (tachy- + Gr. *kardia* heart) excessive rapidity in the action of the heart; the term is usually applied to a heart rate above 100 per minute and may be qualified as atrial, junctional (nodal), or ventricular, and as paroxysmal [Hey95].

**tendon** connective tissue that joins muscle to bone [Maya].

**tensor** an nth-rank tensor (or tensor of order n) in m-space is a mathematical object in m-dimensional space that has n indices and $m^n$ components and obeys certain transformation rules. Each index of a tensor ranges over the number of dimensions of space. However, the dimension of the space is largely irrelevant in most tensor equations (with the notable exception of the contracted Kronecker delta). The notation for a tensor is similar to that of a matrix, except that a tensor $a_{i,j,k,...}$ may have an arbitrary number of indices [Wei].

**thorax** the portion of the anatomy below the neck and above the diaphragm; the chest [Maya].

**thalamus** (L.; Gr. *thalamos* inner chamber) either of two large, ovoid masses, consisting chiefly of grey substance, situated one on each side of and forming part of the lateral wall of the third ventricle. It is divided into two major parts: dorsal and ventral, each of which contains many nuclei [Hey95].

**thrombus** blood clot [Maya].

**tomography** a method of producing a three-dimensional image of the internal structures of a solid object (as the human body) by the observation and recording of the differences in the effects on the passage of waves of energy impinging on those structures - called also stratigraphy [Web02].
transesophageal passing through or performed by way of the *esophagus* [Web02].

transmural 1: passing or administered through an anatomical wall.  
  2: involving the whole thickness of a wall [Web02].

ultrasonography the use of ultrasound as a diagnostic aid. Ultrasound waves are directed at the tissues, and a record is made, as on an oscilloscope, of the waves reflected back through the tissues, which indicate interfaces of different acoustic densities and thus differentiate between solid and cystic structures [Hey95].

valve 1: a structure especially in a vein or lymphatic that closes temporarily a passage or orifice or permits movement of fluid in one direction only  
  2: any of various mechanical devices by which the flow of liquid (as blood) may be started, stopped, or regulated by a movable part that opens, shuts, or partially obstructs one or more ports or passageways; also: the movable part of such a device [Web02].

vein any of the tubular branching vessels that carry blood from the capillaries toward the heart and have thinner walls than the arteries and often valves at intervals to prevent reflux of the blood which flows in a steady stream and is in most cases dark-colored due to the presence of reduced hemoglobin [Web02].

velocity-encoded cine MRI MRI technique for production of (flow) velocity maps [SB99].

ventral corticospinal tract (also anterior corticospinal tract, direct pyramidal tract) a band of nerve fibers that descends in the ventrolateral part of the spinal cord and consists of fibers arising in the motor cortex of the brain on the same side of the body and not crossing over in the decussation of pyramids [Web02].

ventricle a cavity of a bodily part or organ: as  
  1: a chamber of the heart which receives blood from a corresponding atrium and from which blood is forced into the arteries.  
  2: one of the system of communicating cavities in the brain that are continuous with the central canal of the spinal cord, that like it are derived from the medullary canal of the embryo, that are lined with an epithelial ependyma, and that contain a serous fluid.  
  3: a fossa or pouch on each side of the larynx between the false vocal cords above and the true vocal cords below [Web02].

ventricular of, relating to, or being a ventricle especially of the heart or brain [Web02].

ventriculography the act or process of making an X-ray photograph of the ventricle of the heart after injection of a radiopaque substance [Web02].
X ray 1: any of the electromagnetic radiations of the same nature as visible radiation but of an extremely short wavelength less than 100 angstroms that is produced by bombarding a metallic target with fast electrons in vacuum or by transition of atoms to lower energy states and that has the properties of ionizing a gas upon passage through it, of penetrating various thicknesses of all solids, of producing secondary radiations by impinging on material bodies, of acting on photographic films and plates as light does, and of causing fluorescent screens to emit light—called also roentgen ray.

2: a photograph obtained by use of X rays <a chest X ray> [Web02].
Bibliography


Bibliography


[Duk] Duke University Medical Centre. Echo in context - introduction to doppler echocardiography. URL: http://www.echoincontext.com/basicDoppler.asp.


eFunda Inc. efunda - online engineering fundamentals. URL: http://www.efunda.com.


[Fau00] Prof. Richard Faull. Personal communication, 28th September 2000.


[Hei] Andreas Heimdal. Strain rate imaging. URL: http://www.stud.ntnu.no/~heimdal/SRI.


[IIC] IICM - Institute for Information Processing and Computer Supported New Media. Human depth perception. Graz University of Technology, Austria, URL: http://www2.iicm.edu/0x811bc833_0x000dacca.


[Jas97] Vijendra Jaswal. CAVEvis: Distributed real-time visualization of time-varying scalar and vector fields using the cave virtual reality theater.


Kay Medical Group. The structure and function of the heart. URL: http://204.140.221.39/heart.htm.


[KU ] KU Medical Centre. Medical study guide. URL: http://www.kumc.edu/AMA-MSS/study/study.htm.


[MCD+96] Michael J. Moulton, Lawrence L. Creswell, Stephen W. Downing, Ricardo L. Actis, Barna A. Szabo, Michael W. Vannier, and Michael K. Pasque. Spline surface interpolation for calculating 3-D ventricular...


NAG Ltd. The IRIS Explorer homepage. URL: http://www.nag.com/Welcome_IEC.html.


NCSA Visualization and Virtual Environments Group. Training information about IDL. URL: http://www.ncsa.uiuc.edu/SCD/Vis/Software/Idl/Training/Intro/.


[OVS+01] Evren Ozarsland, Baba Vemuri, Xeve Silver, Michelle DeFord, and Tom Mareci. Fiber tract mapping with magnetic resonance diffusion tensor


[Qu02] Danhong (Jessica) Qu. A graphical user interface for colour map design. FoS summer scholarship project report, University of Auckland, February 2002.


Bibliography


Alistair A. Young. Personal communication, 9th December 2002.


