Exploiting Weak Constraints on Object Structure and Appearance for Segmentation of 2–D Images

A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy in the Faculty of Medicine, Dentistry, Nursing and Pharmacy

2001

Mike Rogers
Imaging Science and Biomedical Engineering
Contents

1 Introduction ........................................... 25
   1.1 Diabetic Neuropathy .............................. 26
   1.2 Previous Work ................................... 27
   1.3 Objectives ...................................... 30
   1.4 Data ........................................... 31
       1.4.1 Images ................................... 31
       1.4.2 Annotations ................................. 31
   1.5 Structure of Thesis ............................. 33

2 Image Interpretation Methods ....................... 37
   2.1 Data-Driven Segmentation ...................... 38
       2.1.1 Thresholding ............................... 38
       2.1.2 Edge Detection ............................ 39
       2.1.3 Region Based Methods .................... 40
       2.1.4 Texture Analysis .......................... 40
       2.1.5 Active Contours ........................... 41
       2.1.6 Neural Networks ........................... 42
   2.2 Model Driven Methods .......................... 43
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.1</td>
<td>Modelling Shape</td>
<td>43</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Modelling Appearance</td>
<td>53</td>
</tr>
<tr>
<td>2.3</td>
<td>Model Fitting</td>
<td>55</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Local Refinement</td>
<td>56</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Global Optimisation</td>
<td>57</td>
</tr>
<tr>
<td>2.3.3</td>
<td>Robust Fitting</td>
<td>59</td>
</tr>
<tr>
<td>2.4</td>
<td>Summary</td>
<td>61</td>
</tr>
<tr>
<td>3</td>
<td>Modelling Shape and Appearance</td>
<td>63</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>63</td>
</tr>
<tr>
<td>3.2</td>
<td>Active Shape Models (ASMs)</td>
<td>64</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Modelling Shape</td>
<td>64</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Model Fitting</td>
<td>69</td>
</tr>
<tr>
<td>3.3</td>
<td>Application of ASMs to Capillary Images</td>
<td>72</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Landmarking Strategy</td>
<td>73</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Profile Model Optimisation</td>
<td>75</td>
</tr>
<tr>
<td>3.3.3</td>
<td>Model Parameter Estimation</td>
<td>86</td>
</tr>
<tr>
<td>3.3.4</td>
<td>Evaluation Measure</td>
<td>92</td>
</tr>
<tr>
<td>3.3.5</td>
<td>Evaluation Results</td>
<td>93</td>
</tr>
<tr>
<td>3.3.6</td>
<td>Summary</td>
<td>100</td>
</tr>
<tr>
<td>3.4</td>
<td>Fourier Descriptors</td>
<td>101</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Invariant Fourier Descriptors</td>
<td>103</td>
</tr>
<tr>
<td>3.5</td>
<td>Application of Fourier Descriptors to Capillary Images</td>
<td>110</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Principal Components Analysis of Fourier Parameters</td>
<td>110</td>
</tr>
</tbody>
</table>
3.5.2 Model Fitting ................................................. 112
3.5.3 Evaluation Results ................................. 114
3.5.4 Summary ................................................. 116

3.6 Active Appearance Models (AAMs) ................. 117
3.6.1 Model Fitting .............................................. 119

3.7 Application of AAMs to Capillary Images ............ 120
3.7.1 Optimising Image Warping ....................... 120
3.7.2 Model Fitting .............................................. 125
3.7.3 Summary ................................................. 129

3.8 Summary .................................................. 129

4 Modelling Intermittent Features: Structured Point Distribution and Structured Appearance Models 132
4.1 Introduction ............................................... 132
4.2 Modelling Intermittently Present Data ............... 134
4.3 Data Imputation ........................................... 136
4.3.1 Imputation Methods ................................. 137
4.3.2 Evaluation ............................................... 143
4.4 Modelling Shape and Structure ....................... 149
4.4.1 Shape .................................................... 150
4.4.2 Missing Data Structure ......................... 151
4.4.3 Combined Model ...................................... 154
4.4.4 Calculating Model Parameters .................. 155
4.4.5 Evaluation ............................................... 156
4.5 Modelling Appearance .................................................. 172
  4.5.1 Appearance ......................................................... 173
  4.5.2 Combined Model ................................................... 175
  4.5.3 Evaluation .......................................................... 177
4.6 Search ................................................................. 185
  4.6.1 Structured Active Shape Models (SASMs) ....................... 186
  4.6.2 Structured Active Appearance Models (SAAMs) .............. 192
4.7 Summary ............................................................... 192

5 Modelling Grey-Level Texture 195
  5.1 Texture Discrimination ............................................... 196
    5.1.1 Texture Features ................................................ 197
  5.2 Evaluation ........................................................... 215
    5.2.1 Experimental Design ............................................ 216
    5.2.2 Filter Banks .................................................... 219
    5.2.3 Classification .................................................. 224
    5.2.4 Results and Analysis .......................................... 227
  5.3 Summary ............................................................. 236

6 Nerve Capillary Segmentation 237
  6.1 Experimental Design ............................................... 238
    6.1.1 Annotations .................................................... 240
    6.1.2 Shape and Structure Modelling ................................ 242
    6.1.3 Texture Images ................................................ 245
    6.1.4 Model Fitting .................................................. 248
6.2 Results and Analysis .................................................. 251
  6.2.1 ASM vs SASM – Modelling Constraints ....................... 252
  6.2.2 Wavelet vs Optimal Discrimination Texture Analysis ...... 257
  6.2.3 Automatic Segmentation ......................................... 260
6.3 Discussion ................................................................. 264
6.4 Summary ................................................................. 269

7 Conclusions and Discussion ............................................. 270
  7.1 Modelling Capillary Shape and Appearance ...................... 270
    7.1.1 Model Parameterisation ........................................ 270
    7.1.2 Structured PDMs ................................................ 271
  7.2 Model Fitting .......................................................... 272
  7.3 Texture Analysis ...................................................... 273
  7.4 Future Work and Open Questions .................................... 273
    7.4.1 Modelling ......................................................... 274
    7.4.2 Texture Analysis ................................................ 274
    7.4.3 Boundary Refinement .......................................... 274
    7.4.4 Reproducibility ................................................ 275
  7.5 Overall Performance ................................................ 275
  7.6 Final Statement ....................................................... 277
List of Figures

1.1 Electron microscope images of two endoneurial capillaries, exhibiting a large variation in shape and appearance. Annotations of the basement membrane/endothelial cell material (BMEC) and lumen/endothelial cell material (LEC) boundaries are marked as yellow lines. 27

1.2 Byrne’s pressurised snake boundary measure was the difference between the image texture measure responses from regions one and two. 29

1.3 Examples of nerve capillary image data together with annotations of BMEC and LEC boundaries where available. 33

1.4 Further examples of nerve capillary image data together with annotations of BMEC and LEC boundaries where available. 34

2.1 Gradient ascent search may become trapped in local maxima. Point (a) will not find the global maximum of the fitness function. Point (b) has a better initialisation and will find the true maximum. 57

2.2 Fitting a line model to data using a least squares method is sensitive to outliers. The dotted line results from fitting a line to the data points marked as blue crosses. The solid line represents the idea model fit for this data. 60

3.1 Varying the first three modes from a nerve capillary PDM between $\pm 2\sigma_k$ about the model mean. 68
3.2 Landmarking nerve capillary structures: (a) shows the original annotated BMEC and LEC boundaries, (b) shows the best fitting ellipse to the BMEC boundary together with its major axis. (c) shows the 20 evenly spaced landmark points representing the BMEC boundary starting at the intersection of the BMEC with the major axis of the ellipse. (d) adds the 20 evenly spaced landmark points representing the LEC boundary, starting at the intersect with the major axis, which has been offset to lie through the LEC boundary’s centre of gravity. 76

3.3 Varying the first mode of a typical grey-level profile model associated with a landmark point on the BMEC boundary between $\pm 2\sigma$ of the mode mean. 80

3.4 Varying the first mode of the combined grey-level profile model for the BMEC boundary between $\pm 2\sigma$ of the mode mean. 80

3.5 Example texture images and associated profile data. In each image the BMEC boundary is marked as a yellow line. (a) shows an image with a good intensity edge at the BMEC boundary. (c) shows example profiles across the BMEC boundary from this image. (b) contains many confusing regions close to the BMEC boundary, and (d) shows corresponding profile examples. 82

3.6 ASM search profile model best fit positions are marked as yellow crosses. The green line marks an expert annotation of the image. (a) shows possible best fit outliers circled in red. (b) shows the PDM reconstruction of the fit positions as a red line. 86

3.7 RANSAC model fitting. Profile model best fit positions are shown as yellow crosses, the subset of points used to calculate shape model parameters are shown as green pentagrams and the resulting model fit and consensus set as a red line and circles. 30 of the 50 landmark points were used to calculate the model fit in each case on the same set of model fits. 91

3.8 Distance measures. (a) shortest point to line distance, (b) shortest point to bounded line, as part of a boundary. 92

3.9 A boundary formed by the summation of rotating elliptic phasors, with components shown at three different times, $t_1, t_2$ and $t_3$. $c_1, c_2$ and $c_3$ are the centres of each ellipse. 104
3.10 The smoothing effect of truncating the Fourier summation. Fourier reconstruction of the boundary marked as a dashed black line is shown as a heavy blue line. The Fourier parameterisation used 1, 5, 10 and 50 harmonics shown in (a),(b),(c) and (d) respectively.  

3.11 A comparison ASM and Fourier model fitting. Yellow crosses show profile search best fit positions. The green line shows the PDM instance best fitting the profile search results, and the red line shows the equivalent Fourier model instance. Both models contained 100%, 99.5%, 95% and 66% of total variance in figures (a),(b),(c) and (d) respectively.  

3.12 A Delaunay triangulation of the mean capillary shape.  

3.13 The first four modes of an appearance model built using a piecewise linear warper, based on a Delaunay triangulation.  

3.14 A manual shape-free region with triangulation.  

3.15 The first four modes of an appearance model built using a piecewise linear warper based on a manually defined triangulation.  

3.16 A thin plate spline warp. (a) shows the model mean shape and landmark points, marked as blue crosses. (b) shows the target landmark points that define the thin plate spline. The regular red grid of intermediate points shown in (c) is warped to the grid in (d). Corresponding points in (c) and (d) are used as sample positions for shape-free appearance.  

3.17 The first four modes of an appearance model built using a thin plate spline warper.  

4.1 Two examples from the set of nerve capillary images. Both BMEC and LEC boundaries are marked as yellow lines where available. (a) shows an example where the lumen is visible, (b) is an example where it is not.  

4.2 Diagram representing the strategy used to model intermittently present features.
4.3 The data imputation process for 2D data. A Gaussian model of a 2D data distribution is shown, with mean and covariance marked as a blue cross and ellipse. A partial data vector, in which only the value of variable 1 is known, is shown as a red cross. The value of variable 2 must be imputed to lie somewhere along the dashed line. ........................................... 137

4.4 Data imputation using variable mean for 2D data. The imputed value of variable 2 is shown as a green cross. ................................................................. 138

4.5 Data imputation using PCA for 2D data. The imputed value of variable 2 is shown as a large green cross. Missing data estimates are formed by imputing means, projecting the data onto the first principle component (represented as the dashed diagonal line) and reading off the variable value. ................................................................. 139

4.6 Data imputation using maximum likelihood estimate for 2D data. The maximum likelihood estimate of the value of variable 2 is shown as a large green cross. ................................................................. 140

4.7 Examples from the set of multivariate synthetic data. Variable value is plotted against variable number, showing an imperfect linear relationship in each example. ................................................................. 144

4.8 The Euclidean distance between synthetic data and imputed data. The proportion of data removed before imputation was varied between 1% and 50%. Iterated PCA and ML curves almost coincide. .............. 146

4.9 The Euclidean distance between ideal values of the synthetic data and imputed data. The distance between the ideal data values and the randomly perturbed synthetic data is shown as a horizontal dashed line. Iterated PCA and ML curves almost coincide. .............. 146

4.10 Four examples of BMEC boundary landmark data used to evaluate imputation methods. Landmark points are marked as crosses on each boundary. ................................................................. 147

4.11 The Euclidean distance between nerve capillary landmark data and imputed data. ................................................................. 148

4.12 Difference between the total variance in the nerve capillary landmark data and the imputed data. ................................................................. 148

4.13 Optimal threshold position between two normally distributed classes. 153
4.14 Examples from the set of synthetic training data . . . . . . . . . . . . 158

4.15 Synthetic data model: single mode of variation ±2 std. dev. . . . . 159

4.16 Examples from the set of diabetic nerve capillary training data . . . 160

4.17 First six modes of variation from the diabetic nerve capillary model, all ± 2 std. dev. from the mode mean . . . . . . . . . . . . . . . . . . 161

4.18 Generating invalid shapes from a valid example. (a) shows a valid example of capillary shape. (b) is the shape that results after randomly offsetting each landmark of (a) by up to 10% of the size of the capillary in each dimension. (c) shows the PDM reconstruction of the invalid shape overlayed on the valid shape, (d) shows the same figure for a SPDM reconstruction. . . . . . . . . . . . . . . . . . . . . . . . . . . 162

4.19 Examples from the evaluation of specificity of nerve capillary models. The solid black line shows the original valid training example, the dotted black line shows the randomly perturbed invalid shape. The maximum size of the random offset is given under the corresponding figure as a percentage of capillary size. The solid red line shows the SPDM reconstruction of the randomly perturbed data, and the solid blue line shows the corresponding reconstruction for the PDM. The SPDM reconstructs shapes that are closer to the original valid shape as the maximum size of random offset increases. . . . . . . . . . . . . 164

4.20 Evaluation of the specificity of nerve capillary models, showing the average point–to–point distance of model reconstructions of invalid data to valid shape. The red line shows the distance between the SPDM reconstruction of invalid data and the corresponding valid shape, with error bars of ±1 std. dev. The blue line shows the distance between PDM reconstruction and valid data, also with ±1 std. dev. error bars. SPDM reconstructions of invalid shapes consistently lie closer to valid shape data than PDM reconstructions. . . . . . . . . . . . . . . . . . . . . . . . . . . 165

4.21 Evaluation of the specificity of nerve capillary models, showing the average point–to–point distance of model reconstructions of invalid data. The red line shows distance between SPDM reconstruction and the randomly perturbed, invalid shape data. The blue line shows distance between PDM reconstruction and invalid shape data. Both curves show error bars of ±1 std. dev. PDM reconstructions of invalid shapes consistently lie closer to invalid shape data than SPDM reconstructions. . . . . . . . . . . . . . . . . . . . . . . . . . . 166
4.22 Examples from the set of face shape training data. .......................... 167
4.23 First six modes of variation from the face model, all ± 2 std. dev. from the mode mean ......................................................... 168
4.24 Examples from the evaluation of specificity of face models. The solid black line shows the original valid training example, the dotted black line shows the randomly perturbed invalid shape. The maximum size of the random offset is given under the corresponding figure as a percentage of face size. The solid red line shows the SPDM reconstruction of the randomly perturbed data, and the solid blue line shows the corresponding reconstruction for the PDM. The SPDM reconstructs shapes that are closer to the original valid shape as the maximum size of random offset increases. ......................................................... 169
4.25 Evaluation of the specificity of face models, showing the average point-to-point distance of model reconstructions of invalid data to valid shape. The red line shows the distance between the SPDM reconstruction of invalid data and the corresponding valid shape, with error bars of ±1 std. dev. The blue line shows the distance between PDM reconstruction and valid data, also with ±1 std. dev. error bars. SPDM reconstructions of invalid shapes consistently lie closer to valid shape data than PDM reconstructions. ......................................................... 170
4.26 Evaluation of the specificity of face models, showing the average point-to-point distance of model reconstructions of invalid data. The red line shows distance between SPDM reconstruction and the randomly perturbed, invalid shape data. The blue line shows distance between PDM reconstruction and invalid shape data. PDM reconstructions of invalid shapes consistently lie closer to invalid shape data than SPDM reconstructions. ......................................................... 171
4.27 Modelling strategy for shape and appearance information. .......... 173
4.28 Shape–Free representations of face appearance data. Blue crosses and red circles represent sample points. (a) shows sample points in the shape–free representation of face data with no moustache, (b) shows corresponding sample positions after warping to a face shape in the training set with no moustache marked. (c) shows sample points in the shape-free representation with points within the moustache structure circled in red, (d) shows the corresponding warped sample positions in a face shape in the training set. ......................................................... 176
4.29 Examples from the set of face appearance training data. 178

4.30 First six modes of variation from the face structured appearance model, all ± 2 std. dev. from the mode mean. 179

4.31 First six modes of variation from the face standard appearance model, all ± 2 std. dev. from the mode mean. 180

4.32 Examples from the set of capillary appearance training data. 182

4.33 The first six modes of variation from the nerve capillary structured appearance model, all ± 2 std. dev. from the mode mean. The LEC boundary is marked when it is present in the model instance. 183

4.34 The first six modes of variation from the nerve capillary appearance model, all ± 2 std. dev. from the mode mean. 184

4.35 SASM fitting results: (a) and (c) show model starting positions in green, (b) and (d) show the corresponding final model fits after 5 iterations. 190

4.36 Model fitting robustness comparison. Mean point–to–point distance between model fit position and annotated landmarks for a set of image searches. Model searches were initialised at randomly offset annotation data. The size of the maximum offset was varied between 1% and 25% of total annotation size. ASM search results are shown as a blue line, and SASM are shown in red. 191

4.37 SAAM fitting results for two training images: (a) and (d) show original images, (b) and (e) show model initialisation each with the moustache feature incorrectly represented, and (c) and (f) show final model fits after five iterations, showing the correct state of the moustache. 193

5.1 Partitioning the Fourier power spectrum into bins. Axis labels are vertical horizontal spatial frequencies. 200

5.2 Laws texture filters $L_5$, $S_5$, $E_5$, $R_5$ and $W_5$. 202

5.3 The Fourier domain partitioned by the 25 Laws texture energy measures. Axis labels are vertical and horizontal normalised spatial frequencies. 203
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4</td>
<td>An illustration of the partitioning of the frequency domain using dyadic Gabor filtering. Each filter is represented by one centre–symmetric pair of lobes. Axes are horizontal and vertical spatial frequency.</td>
</tr>
<tr>
<td>5.5</td>
<td>The discrete wavelet transform implemented by low-pass ($H_0$) and high-pass ($H_1$) filters with down-sampling ($\downarrow 2$).</td>
</tr>
<tr>
<td>5.6</td>
<td>Frequency domain partitioned by a three-level dyadic wavelet transform.</td>
</tr>
<tr>
<td>5.7</td>
<td>Frequency domain partitioned by a three-level non-dyadic wavelet transform.</td>
</tr>
<tr>
<td>5.8</td>
<td>A dataflow diagram illustrating the filter-bank statistical texture analysis process.</td>
</tr>
<tr>
<td>5.9</td>
<td>Images used in the texture discrimination evaluation. BMEC boundary structure is marked as a yellow line.</td>
</tr>
<tr>
<td>5.10</td>
<td>Biorthogonal wavelet high-pass and low-pass filters used for texture feature generation.</td>
</tr>
<tr>
<td>5.11</td>
<td><em>Optimal discrimination</em> image filters generated from the textures inside and outside of the BMEC boundary.</td>
</tr>
<tr>
<td>5.12</td>
<td>Sampling region and 100 random class samples for an example image. Inside BMEC boundary samples are shown as red crosses, outer samples are shown in green.</td>
</tr>
<tr>
<td>5.13</td>
<td>Texture images generated using wavelet texture features with different classifiers.</td>
</tr>
<tr>
<td>5.14</td>
<td>Texture images generated using Laws texture feature and wavelet Bior3.5 texture features with linear classification. (a),(c) and (e) were generated using Laws texture features, (b),(d) and (f) used wavelet texture features.</td>
</tr>
<tr>
<td>5.15</td>
<td>Texture images generated using Laws texture feature and optimal discrimination texture features. (a),(c) and (e) were generated using Laws texture features with a linear classifier, (b),(d) and (f) used optimal discrimination texture features with a mixture model classifier representing each class with eight kernels.</td>
</tr>
<tr>
<td>5.16</td>
<td>Histogram of model fit to annotation distances for images created using Laws texture features with a linear classifier.</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>5.17</td>
<td>Histogram of model fit to annotation distances for images created using biorthogonal wavelet Bior3.5 texture features with a linear classifier.</td>
</tr>
<tr>
<td>5.18</td>
<td>Histogram of model fit to annotation distances for images created using optimal discrimination texture features with a linear classifier.</td>
</tr>
<tr>
<td>6.1</td>
<td>Histogram of point-to-boundary distances for annotations of the same image. (a) entire histogram, (b) tail of the distribution.</td>
</tr>
<tr>
<td>6.2</td>
<td>Example annotated BMEC boundaries exhibiting ambiguous interpretations. ( \mu_{im} ) is the average point-to-boundary distance between each annotation point and the nearest different annotation boundary for each image. (a) shows the most ambiguous set of annotations, (b) is the most consistent. (c) and (d) are examples with ( \mu_{im} ) similar to the overall average ( \mu_{all} ).</td>
</tr>
<tr>
<td>6.3</td>
<td>Examples from the set of texture images generated using a linear classification of biorthogonal wavelet texture measures.</td>
</tr>
<tr>
<td>6.4</td>
<td>Examples from the set of texture images generated using a mixture model classification of optimal discrimination texture measures.</td>
</tr>
<tr>
<td>6.5</td>
<td>Fitting an incorrectly oriented shape model to image data.</td>
</tr>
<tr>
<td>6.6</td>
<td>Histogram of ASM fit to annotation point-to-boundary distances. (a) shows the entire distribution, (b) shows only the tail of the histogram.</td>
</tr>
<tr>
<td>6.7</td>
<td>Histogram of SASM fit to annotation point-to-boundary distances. (a) shows the entire distribution, (b) shows only the tail of the histogram.</td>
</tr>
<tr>
<td>6.8</td>
<td>Example segmentation results. Final model fit positions are shown in red, with all BMEC boundary annotations marked as yellow lines. (a), (c) and (e) show ASM search results, (b), (d) and (f) show corresponding SASM results.</td>
</tr>
<tr>
<td>6.9</td>
<td>Histogram of SASM fit to annotation point-to-boundary distances for wavelet texture images. (a) shows the entire distribution, (b) shows only the tail of the histogram.</td>
</tr>
<tr>
<td>6.10</td>
<td>Histogram of SASM fit to annotation point-to-boundary distances for optimal discrimination texture images. (a) shows the entire distribution, (b) shows only the tail of the histogram.</td>
</tr>
</tbody>
</table>
6.11 Example wavelet and optimal discrimination texture image SASM segmentation results. Final model fits are shown in red, with annotations marked as yellow lines. (a), (c) and (e) show SASM search results on wavelet texture images, (b), (d) and (f) show SASM results on optimal discrimination texture images. 261

6.12 Histogram of multi-resolution SASM fit to annotation distances. 262

6.13 Histogram of genetic algorithm SASM fit to annotation distances. 262

6.14 Example multi-resolution and genetic algorithm SASM segmentation results. Final model fits are shown in red, with annotations marked as yellow lines. (a), (c) and (e) show multi-resolution SASM search results on wavelet texture images, (b), (d) and (f) show genetic algorithm SASM results on wavelet texture images. 263

6.15 Histogram of mean SASM segmentation to annotation distances for wavelet texture images. 265

6.16 Histogram of mean SASM segmentation to annotation distances for optimal discrimination texture images. 265

7.1 Examples of segmentations where $\mu_{im} < 10$ pixels. 276
List of Tables

3.1 Comparison of model fitting performance using standard and normalised grey-level profile models. \( \mu_{all} \) and \( \sigma_{all} \) are the mean and standard deviation of the point-to-boundary distances between each model fit landmark and the nearest annotated boundary respectively, \( \mu_{acc} \) is the average of the smallest 25% of distances, giving a measure of the accuracy of successfully located model points. \( \mu_{rob} \) average of the largest 25% of distances, giving a measure of the size of the largest failure, and hence model fitting robustness. 94

3.2 Comparison of model fitting performance using normalised and combined grey-level profile models. 95

3.3 Comparison of model fitting performance using selected data to train grey-level profile models. The first row shows results using unselected data to train profile models. The second shows results using only selected intensity edge data, and the third shows results achieved by explicitly modelling locally confusing evidence together with intensity edge information using a mixture model. 96

3.4 Effect of varying profile model patch dimensions on ASM model fitting performance. A normalised, combined profile model trained with selected intensity edge data was used. 97

3.5 Effect of varying the total amount of variance represented in profile models on model fitting performance. Normalised, combined profile models representing selected training data with patch dimensions of 5 \( \times \) 40 were used. 98

3.6 Model fitting performance using weighted pose estimation. 99
3.7 RANSAC Model fitting performance. The first row shows results using standard model fitting. The second row shows RANSAC model fitting results using profile models containing selected intensity edge data. The third row used RANSAC fitting with profile models representing both intensity edge and locally confusing appearance data.

3.8 Effect of varying Fourier model building parameters on reconstruction error and number of modes. Reconstruction error is defined as the mean point to line distance between 50 evenly spaced points on the reconstructed boundary and the original annotation.

3.9 Effect of varying PDM building parameters on reconstruction error and number of modes. Reconstruction error is defined as the mean point to line distance between the landmarks on the reconstructed boundary and the original annotation.

3.10 Fourier Descriptor Model fitting performance. $\mu_{all}$ and $\sigma_{all}$ are the mean and standard deviation of the point-to-boundary distances between each model fit landmark and the nearest annotated boundary respectively, $\mu_{acc}$ is the average of the smallest 25% of distances, giving a measure of the accuracy of successfully located model points. $\mu_{rob}$ average of the largest 25% of distances, giving a measure of the size of the largest failure, and hence model fitting robustness.

3.11 Comparison of AAM fitting performance using three different warper types. $\mu_{all}$ and $\sigma_{all}$ are the mean and standard deviation of the point-to-boundary distances between each model fit landmark and the nearest annotated boundary respectively, $\mu_{acc}$ is the average of the smallest 25% of distances, giving a measure of the accuracy of successfully located model points. $\mu_{rob}$ average of the largest 25% of distances, giving a measure of the size of the largest failure, and hence model fitting robustness.

5.1 Cross-validation mean misclassification rates for each classifier and each set of texture features for 10 randomly chosen images. The standard deviations of the misclassification rates across the set of images are shown in brackets.
5.2 Model fitting results for texture measures. (a) shows results for manual texture features, (b) for Laws, (c) for Bior1.5 wavelets, (d) for Bior3.5 wavelets and (e) for optimal discrimination filtering. $\mu_{all}, \sigma_{all}$ are mean and std. dev. of all point–to–boundary distances between final model positions and each annotation. $\mu_{acc}$ is average distance for the best 25% of values. $\mu_{rob}$ is average distance the worst 25% of distances.

6.1 Summary of point–to–boundary distances for annotations of the same image. $\mu_{all}, \sigma_{all}$ are mean and std. dev. of all annotation to annotation distances. $\mu_{acc}$ is mean of the closest 25% of measurements. $\mu_{rob}$ is mean distance for the worst 25%.

6.2 Model fit to annotation point–to–boundary distance summary, for ASM and SASM searches. $\mu_{all}, \sigma_{all}$ are mean and std. dev. of all model fit to annotation distances. $\mu_{acc}$ is mean model fit to annotation distance of the best 25% of model fit points. $\mu_{rob}$ is mean distance for the worst 25% of model fit points.

6.3 Model fit to annotation point–to–boundary distance summary, for SASM searches of wavelet and optimal discrimination images.

6.4 Model fit to annotation point–to–boundary distance summary, for SASM searches of wavelet texture images using single resolution, multi–resolution and genetic algorithm search.
Abstract

Prior knowledge can be used to constrain complex image interpretation tasks. When objects are highly variable, the constraints derived from the prior knowledge will be weak. This thesis investigates how best to maximise and utilise weak constraints, using as an example the location of boundaries (segmentation) in nerve capillary images.

Capillaries imaged by electron microscopy show a complex textured appearance, making segmentation difficult. Considerable variation occurs among boundaries manually positioned by human experts. Previous attempts to detect boundaries using a data-driven approach have proved unsuccessful due to the existence of confusing image evidence in the vicinity of these boundaries.

Point Distribution Models (PDMs) are found to be the most appropriate modelling method, despite the fact that the capillaries have no identifying landmarks. An extension to PDMs called Structured PDMs is proposed, which is designed to represent the intermittently present lumen structure. Including this feature results in a more specific model. A comparison of texture analysis schemes is carried out. Results show that wavelet texture analysis gives the best classification of capillary structures.

Finally, a quantitative evaluation shows that model driven methods improve on the accuracy and robustness of data-driven segmentation performance. More specifically, SPDMs are shown to significantly improve segmentation results compared to PDMs. Genetic search provides a promising method of fully automatic segmentation.
Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.
Copyright

1. Copyright in text of this thesis rests with the Author. Copies (by any process) either in full, or of extracts, may be made only in accordance with instructions given by the Author and lodged in the John Rylands University Library of Manchester. Details may be obtained from the Librarian. This page must form part of any such copies made. Further copies (by any process) of copies made in accordance with such instructions may not be made without permission (in writing) of the Author.

2. The ownership of any intellectual property rights which may be described in this thesis is vested in the University of Manchester, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of the University, which will prescribe the terms and conditions of any such agreement.

Further information on the conditions under which disclosures and exploitation may take place is available from the Head of the Division of Imaging Science and Biomedical Engineering.
I would like to thank the following people for their help during my research:

My supervisor Dr Jim Graham for his guidance and support during my research and the period spent writing this thesis. Dr R A Malik and Dave Walker for help annotating the images used in this thesis.

Fellow students and ISBE staff, in particular: Dr Kevin Walker, Steve Caulkin, Chris Wolstenholme, Dr Carol Twinning, Dr Gavin Wheeler and Dr Gareth Edwards for many productive technical discussions and not so productive evenings out. Also Dr Tim Cootes and Prof Chris Taylor, who were always approachable when an extra source of advice was needed.

Warren Mittoo and the rest of the support staff for keeping all the computers up and running.

My house mates Donna, Maura and Mel, and most of all my girlfriend Kate, who all looked after me with great patience and understanding during the writing of this thesis. I am eternally grateful.

Thanks also to my Mum and Dad and the rest of my family who have encouraged me to continue my education.

Finally, I would like to thank Brian, my cat, who has been a constant source of companionship.
About the Author

Mike Rogers received a first class BSc in artificial intelligence from Manchester University in 1996. In 1997 he received an MSc in cognitive science, also from the University of Manchester. In October 1997, he commenced research in the Department of Medical Biophysics (now Imaging Science and Biomedical Engineering) again at the same institute, for which this thesis is submitted. During this period he published the following papers related to the work in this thesis.


Chapter 1

Introduction

In general, analysis of non-trivial objects in images requires the use of some a\textit{pri}ori knowledge. Frequently, this knowledge is couched as a statistical model of the variance of a particular object feature. These models, when fitted to data, provide estimates of the likelihood of the data given the model and therefore can be used to constrain image interpretation.

In many cases, the variation present in the feature being modelled is large. The constraints imposed by such a model are therefore weaker than those of a model of a more consistent feature. When this is the case, it is necessary to maximise the constraints applied by choosing an appropriate modelling paradigm and maximising the information that it encapsulates. In 2-D images, information such as shape, appearance and structure are all valuable sources of constraint on image interpretation and all should be integrated to maximise prior information.

Objects with wide variation in appearance and shape are frequently found in medical images. The benefits of using prior knowledge have been demonstrated many times in the case of medical images, where shapes, while variable, are fairly well-defined \cite{22,32,31,79,132,133,142}. One example where shape and appearance variability is much greater than usual is nerve capillary images. In this case, the constraining
effects of modelling will be especially weak. This work uses the segmentation of nerve capillary images as an exemplar of how to use weak \textit{a-priori} information to improve robustness and accuracy of image interpretation.

1.1 Diabetic Neuropathy

Peripheral neuropathy is an important and debilitating symptom of diabetes. Among the pathological manifestations of neuropathy is microangiopathy (disease of small blood vessels) which affects the capillaries in the endoneurium - the interstitial connective tissue in peripheral nerves separating individual nerve fibres. The two main effects are:

- a thickening of the basement membrane or an accumulation of basement membrane material in the capillaries causing an apparent contraction of the luminal area, and

- a proliferation of endothelial cell material together with a thickening of the basement membrane, which manifests itself in arteries, arterioles and occasionally venules.

The aetiology of the condition is unknown. Quantitative studies of the variation in shape and size of the Basement Membrane (BM), Endothelial Cell (EC) and lumen region may shed light on the progression of nerve capillary damage \cite{89}\cite{90}. The structure of two endoneurial capillaries as they appear in electron micrographs is shown in figure 1.1.

Currently segmentations of these areas for the purpose of measurement are taken by hand \cite{89}, which is a time consuming process and restricts the quantity of samples that can be analysed. There is a requirement for an automated approach, both to reduce the labour required and to make the measurements more objective.
Chapter 1. Introduction

Figure 1.1: Electron microscope images of two endoneurial capillaries, exhibiting a large variation in shape and appearance. Annotations of the basement membrane/endothelial cell material (BMEC) and lumen/endothelial cell material (LEC) boundaries are marked as yellow lines.

The nerve capillary structures have a complex appearance, containing no locally consistent anatomical features and wide variation in shape, structure and appearance. The image evidence defining the required boundary is extremely variable from image to image and is often highly ambiguous, containing many areas of confusing local image evidence. Manual expert annotations show considerable differences, even when obtained from the same individual at different times. The image acquisition process (section 1.4) can lead to images of differing quality, further adding to the natural variability of the capillaries themselves. The images presented in figure 1.3 are examples from the set of data used in this work.

1.2 Previous Work

Byrne [18] presented a method of segmenting nerve capillary images at the basement membrane/endothelial cell material (BMEC) boundary. He used active contour models (snakes) [76], driven by an image force based on textural information. Both pressurised [27] and region based [74] snakes were assessed. Areas of locally misleading
image evidence are often present around the BMEC boundary. To reduce the effect of this confusing data, paired snakes [61] were used as a method of combining two independent assessments of the available information. The snakes are attached with a spring force, so they move across the image towards one another until a suitably strong image force is encountered. Both snakes were manually initialised for each segmentation. One snake was started from within the desired boundary and the other snake from the outside, the final segmentation being taken as the mean position of the two snake results.

The image force for the snakes used a weighted linear combination of texture measures to discriminate between regions. The texture measures used were: local average luminance, gradient [140], smoothness, entropy and seven combinations of Laws [84][82][83] texture filters. The weight for each measure was determined by a linear discriminant analysis between samples from the basement membrane and endothelial cell regions. For the pressurised snakes, a measure of boundary strength was used as the image force. The boundary strength was obtained from the difference between the region responses on either side of a candidate boundary, as shown in figure 1.2. At a true boundary, this measure is maximised. For region snakes, the snake pressure term is linked to the statistics of the texture measures within the manually initialised boundary. A binary goodness function was used. This was set to +1 for pixels with image measures within the range specified by values within the initial region and -1 for those with measures outside this range. When a snake element encounters a pixel outside the range, the direction of expansion at that element is reversed. In this way, the snake advances across the image until it encounters pixels with image measures that differ significantly from those in the initial region. These pixels should correspond to a boundary between structures, in this case the BMEC boundary.

Byrne used dynamic programming as a search technique to refine the snake segmentations. This was done to obtain a higher resolution boundary that was more closely matched to the image data, removing the effects of the internal smoothness constraint of the snake segmentations.
Chapter 1. Introduction

Figure 1.2: Byrne’s pressurised snake boundary measure was the difference between the image texture measure responses from regions one and two.

The final boundary positions were evaluated by comparison with boundaries placed manually by a clinical expert. The delineation of structures in nerve capillary images is often unclear, and there is considerable scope for variation in manually placed boundaries. The snake segmentations were considered acceptable if an expert judged that their position was as good as any other, even if he would have positioned it slightly differently in the absence of any prompting. This was achieved by presenting a clinician with the snake results in an interactive tool that allowed any unsatisfactory points to be adjusted. Evaluating the segmentations in this way introduces a bias towards accepting the suggested automatic segmentation because of the effort required to move many points a small distance. A more objective way to evaluate distance from an ambiguous boundary would be to obtain a set of independent annotations for each image so as to gain a measure of uncertainty in the results, as has been done in this work (see section 1.4).

Byrne carried out segmentations on the same set of images that we have used in this work (see section 1.4). Paired statistical region based snakes were found to result in greatly superior segmentations than those of single or paired pressurised snakes, both in terms of accuracy and robustness. The best results showed that the average
distance each automatically generated boundary point was moved was 16.91 pixels, with a standard deviation of 17.09. The results showed a wide range of erroneous boundary positions, indicating a fundamental lack of robustness in the technique. This lack of robustness was largely ascribed to the presence of confusing local image evidence close to the BMEC boundary and the variability in appearance and shape of the capillary structures.

Byrne’s methods, with the exception of the linearly modelled textural image force, were entirely data-driven. No application specific \textit{a-priori} information was used to constrain image segmentation. The nerve capillary images are extremely complex and highly variable with confusing areas of local evidence, so it is perhaps not surprising that neglecting to utilise all available sources of constraint, resulted in a severe lack of robustness and accuracy.

1.3 Objectives

The aim of this work is to apply constraints from \textit{a-priori} information to the segmentation of nerve capillary images, with the goal of improving robustness and accuracy of Byrne’s [18] data-driven methods. Despite the highly variable and ambiguous nature of the data, there are potential constraints to apply to the segmentation problem:

- shape – the capillaries are variable in shape, but not infinitely so,
- structure – the lumen is always inside the BMEC boundary, although its position is variable and it is often missing,
- appearance – there are differences in texture between the various structures, although appearance is highly variable across the set of the images.

We will investigate methods of exploiting each area to improve the segmentations of capillary images at the BMEC boundary.
Constraints such as these have been applied in many areas of image analysis. Handcrafted and statistical models have been successfully used to increase robustness and accuracy in medical image analysis [22][25][32][31][50][79][132][133][142], and other areas of machine vision [9][46][60][125]. There is much greater variation in shape and appearance and far less consistent anatomical structure present in capillary images than in the vast majority of instances where these techniques have been applied. Despite this, these modelling techniques can still be useful in constraining image segmentation, but must result in comparatively weak constraints. This work investigates how these weak constraints can be imposed using statistical models with the aim of improving robustness and reducing the effect of locally confusing image evidence.

1.4 Data

1.4.1 Images

The electron microscope images used in this work were obtained from biopsies of human nerve tissue [18]. The images have been digitised to 256 level grey-scale images of size 768 × 575 pixels. In total 56 images are available for study. However, only a subset of 33 of these images have been retained for analysis as the remainder have been judged by clinicians to be of too poor quality to be used. Figure 1.3 shows eight examples from our image set. One of the images (figure 1.3(c)) is a poor quality rejected image, the other seven have been accepted for use. Figure 1.3(d) exhibits extremely atypical appearance.

1.4.2 Annotations

The structures in the nerve capillary images are highly complex and potentially ambiguous. For this reason, we have obtained multiple annotations, up to four for each image. Images marked less than four times have been declared unusable one or more
times. Two annotation processes have been used to obtain this data. The first was used by Byrne [18] to evaluate the performance of paired snakes at detecting the BMEC boundary. The snake results were presented to a clinician, who modified unacceptable boundary segments using an interactive computer program. A single annotation for each image was obtained in this way. In the second method, the clinician was presented with a clean printout of each image and asked to draw the appropriate boundaries in coloured ink. The annotation was then extracted from the scanned images by a semi-automatic process of line detection and realignment and scaling. Each annotation acquired in this way was manually checked. Three sets of annotations were produced in this way. Two sets were produced by one clinician on separate occasions and the other by a different clinician. In total, up to four annotations are available for each image.

Figures 1.3 and 1.4 show 10 examples of the images that were presented for expert annotation on three separate occasions. The BMEC and lumen/endothelial cell material (LEC) boundaries are marked on the images where they were delineated by the clinicians, together with the adjusted snake boundary positions. Nine of the images have been annotated with the BMEC and LEC at least once. The tenth image was rejected as too poor to be of use by each expert, and the snake result boundary position was rejected by Byrne (figure 1.3(c)). It is important to note that, in each of the images where multiple annotations have been made, there are significant differences in interpretation for the boundary positions. Most commonly, this occurs in local regions of the boundary where there is a high level of ambiguity in the image (figure 1.4(d), 1.4(b)). Occasionally, an annotation of the BMEC boundary has been marked some distance away from the others around the majority of the boundary, indicating an alternative interpretation of the anatomical position of the boundary (1.3(d),1.4(c)). In some cases, there is a widely varying interpretation of the position and extent of the lumen (1.4(b),1.4(c)). This is most common when the lumen is highly constricted. The variation in the positions of annotations seen in this set of eight images is typical of that found across the entire image set.
Chapter 1. Introduction

Figure 1.3: Examples of nerve capillary image data together with annotations of BMEC and LEC boundaries where available.

1.5 Structure of Thesis

In the remainder of this dissertation, we examine and develop methods to add constraints to the segmentation of nerve capillary structures and present an evaluation of the accuracy and robustness of our chosen solution. The structure of the thesis is as follows.

Chapter 2: Image Interpretation Methods

This chapter give a review of methods that may be used for image segmentation and interpretation. We give a review of data–driven approaches, but focus on model–based methods that can be used to impose constraints on image interpretation.
Figure 1.4: Further examples of nerve capillary image data together with annotations of BMEC and LEC boundaries where available.
Chapter 3: Modelling Shape and Appearance

This chapter gives a detailed description of three object modelling techniques, namely Active Shape Models (ASMs), Fourier Descriptors and Active Appearance Models. We investigate the suitability of each method for application to the nerve capillary structure in our images and find that the Active Shape Models provide the best modelling performance. We describe experiments to determine optimal values for the various parameters of the ASM when used for interpretation of our images. We also evaluate several model fitting algorithms as alternatives to the standard ASM technique, including a weighted scheme and Random Sample Consensus (RANSAC) fitting.

Chapter 4: Modelling Intermittent Features: Structured Active Shape and Structured Appearance Models

In this chapter, we introduce a novel method of modelling features that are only intermittently exhibited. This has direct application for the analysis of diabetic nerve capillaries, as the lumen can become so constricted as to be practically unobservable. The method is based on the imputation of missing data, and we propose a novel method to achieve this whilst minimising any bias introduced to further statistical analyses. We support our design with a comparative evaluation of our technique against those presented in the literature.

The technique is presented as an extension to ASMs and AAMs, and adds a component to model the structure of the missing data, hence the name Structured Active Shape Models. We present evaluations of the models on both capillary images and face images, as well as on synthetic data.

Chapter 5: Modelling Grey-Level Texture

This chapter provides a very brief review of textural analysis methods. We present investigations of the applicability of several texture analyses using techniques including Laws texture filters, wavelets and “optimal filter” design. The various images
produced by the methods are compared in terms of their discrimination potential around the BMEC boundary and their model fitting performance.

**Chapter 6: Nerve Capillary Segmentation**

In this chapter, we integrate the chosen techniques from chapters 3, 4 and 5 to segment capillary images at the BMEC boundary. Our method is automatic and uses both global and local search techniques to optimise performance. Results indicate that the method is not robust enough to be used directly in medical research. However, significant improvements are made over a data-driven segmentation scheme from a previous study, showing that utilising model constraints improves performance.

**Chapter 7. Conclusion and Discussion**

This chapter summarises the various techniques and results presented in the previous chapters. We describe how the work presented is more widely applicable than only to capillary images and propose alternative application areas. We also give specific recommendations for future work on the segmentation of nerve capillary images.
Chapter 2

Image Interpretation Methods

This chapter presents a brief review of image interpretation techniques that have the potential to be applied to the segmentation of nerve capillary images. Firstly, we give a review of data driven segmentation algorithms. Data driven segmentation is sensitive to image data complexity and ambiguity. Adding prior knowledge to a segmentation scheme is known to improve segmentation results. In section 2.2, we give a description of various methods of representing prior knowledge that can be used to impose constraints on segmentation. Many of the most recent techniques involve the creation of a statistical model of shape and appearance. In each case, we consider the ability of the technique to represent information about capillary structures. To provide segmentations, statistical models must be fitted to image data. Section 2.3 gives an overview of the various techniques that have been applied to this problem.

In section 2.4, we give a summary of the methods that show promise for application to nerve capillary images.
2.1 Data–Driven Segmentation

Data–driven segmentation methods use image values and heuristic information about objects. Heuristic information is usually in the form of assumptions about boundary information or region homogeneity. Data driven techniques must be carefully tailored to the specific segmentation task. The methods are usually of limited use when simple homogeneity conditions are not easily met, due to image data complexity or ambiguity.

2.1.1 Thresholding

One of the most computationally simplistic methods of image segmentation is thresholding. A brightness contrast threshold is used to segment objects and background. Thresholding relies on the fact that many objects are characterised by a roughly constant or linearly varying intensity values over their entire area. Various methods of thresholding have been developed. Many rely on the presence of a bimodal grey–level histogram. Histogram transformation [67][148] and optimal thresholding are methods used to determine threshold values in the absence of a bimodal image histogram.

Histogram transformation methods [67][148] separate pixels from regions of high image gradient from other pixels to form histograms of border regions between objects or background regions only. Optimal thresholding methods approximate the grey–level histogram with two or more probability density functions. Thresholds are selected at positions that cause the minimum probability of misclassification. A survey of thresholding methods performed by Sahoo et al [120] showed the optimal thresholding method of Otsu [108] to be one of the better methods, showing good uniformity and accurate representation of shape. As statistical methods are used to calculate the optimum threshold, some measure of the quality of the segmentation is also available.

Thresholding, whether arbitrary or optimally designed from image statistics is not
useful in cases where comparison of intensity values leads to a poor discrimination or where there is little region homogeneity in terms of raw pixel values. This can be caused by similar grey-level histogram distributions for objects in the images, or where a large variation in grey-level appearance is found across the set of objects. Both these features are exhibited by nerve capillary images, therefore thresholding directly on image values would be a poor choice of segmentation algorithm.

2.1.2 Edge Detection

Intensity edges are often regarded as the most important image feature to guide segmentation. This is due to their close association with structural changes in images. There are many methods of detecting raw edges, one of the most popular was formulated by Canny [20]. His method is optimal with respect to finding step edges corrupted by white noise. The technique involves filtering the image with a Gaussian smoothing filter with standard deviation $\sigma$. Local edge normal directions are estimated from the smoothed image, and edge locations determined using non-maximal suppression. The resulting image can be thresholded by edge magnitude to produce a binary image of raw edge fragments. Another common method of edge detection is the Laplacian of Gaussian or Mexican Hat filter [94]. This filter can be used to locate zero-crossings in the second derivative of image gradient. Zero-crossings correspond to extrema in the first derivative, which result from edges in the image. Unlike Canny’s method, the Laplacian of Gaussian operator does not calculate edge direction, only magnitude. A comparison of edge detection approaches and performance can be found in [43].

Further processing is required to produce coherent boundary representations from raw edge fragments. Usually edge detection images are thresholded, often with hysteresis, and a relaxation process is applied. Boundary representations can be extracted from these images. Techniques to do this include heuristic graph searching [106] and dynamic programming [8]. Both are concerned with finding the optimal route through
a set of nodes with associated costs. The generalised Hough transform [4] can be used to find shapes in binary edge [96] or grey-level [129] images. This can be used to add information about the expected shape of the desired boundary; however, this method does not cope at all well with shape deformations and so is only useful if the shape of the object to be segmented is known a-priori and fixed. As the basement membrane boundary in the nerve capillary images cannot be clearly identified using edge information alone, edge based segmentation techniques are not useful.

2.1.3 Region Based Methods

Segmentation can also be achieved using properties of the regions. The basic technique is to group pixels based on region homogeneity criteria. A classic form of region based segmentation is the split and merge technique [72]. The idea is to divide an image into smaller regions recursively so that each region satisfies a homogeneity criterion. Regions that do not satisfy the measure are divided, and adjacent regions that are homogeneous are merged. Region homogeneity is defined in terms of the pixel properties within regions. Pixel properties can be quantified using texture measures, of which there are many. This is a potentially useful method for application to nerve capillary images. However, the variation in the appearance of regions to be segmented throughout the image set make it extremely difficult to design a homogeneity measure that is applicable across the entire image set. The presence of locally confusing image evidence would also lead to inaccurate segmentations even if such a measure could be devised.

2.1.4 Texture Analysis

The various structures in capillary images are not characterised by homogeneous grey-level image values. Boundaries between the structures are not consistently associated with high edge magnitude. However, capillary regions are characterised by
textured appearance. Although this appearance is variable from image to image, texture analysis offers a promising pre-processing step to facilitate other segmentation algorithms.

The field of texture analysis is extremely wide and varied. In chapter 5 we describe an investigation into the textural properties of capillary images. This chapter contains a full description of various texture discrimination techniques that we have considered.

### 2.1.5 Active Contours

Kass et al [76] proposed a method of using image features to drive an energy minimising spline curve called an active contour model, or snake. The snake takes the form of a continuous curve, which dynamically deforms itself from a starting position to fit an arbitrarily defined image feature. Snakes introduce some a-priori information in the form of an overall smoothness constraint. However this constraint is not based on any application specific information about the shape or appearance of the target region. Therefore the Active Contour Model does not constitute a model-based method.

Extensions of the method exist that can add additional task specific constraints to deformations [3][54][150]. These extra constraints allow more direct control of the snake’s behaviour. One possible constraint would be to restrict the minimum distance between adjacent points on the curve. This would prevent points from bunching together around regions of the strongest image strength. Cohen [27] introduced an inflationary term to the snake formulation. This causes the contour to act like a balloon, expanding through local, weak image evidence and only becoming caught on areas of strong image evidence.

Various other snake formulations exist, including region based snakes called Active Region Models [74] and paired snakes, called Dual Contour Models [61]. Active region models link the inflationary term of the snake to image data from the region enclosed
by the snake, forming a segmentation based on region homogeneity. Dual contour models attempt to counteract the effects of areas of local image evidence by using two snakes. One snake starts within the desired feature and the other snake outside. The snakes are connected with spring forces, which cause them to be attracted to one another, as well as to image features. The mean position of the snakes is taken as the segmentation.

Snakes rely on the formulation of an image evidence energy function to achieve a segmentation. In Byrne’s study [18], an energy function based on a texture measure which distinguishes between the regions of interest in the nerve capillaries, was used to drive various paired snakes to affect a segmentation at the BMEC boundary.

2.1.6 Neural Networks

Neural networks can be used to segment images. There is an element of modelling in such techniques as implicit models of solutions are generated within the weights of the networks. Campbell and Thomas [19] used Kohonen nets to segment natural images in an unsupervised manner. Images were preprocessed using Gabor texture filters and the nodes of the Kohonen network were labelled with the class that is most often projected onto it from a set of labelled images.

Poli and Valli [113] used Hopfield neural networks to segment X-ray images and MR images of the brain. The segmentation task was formulated as a set of requirements. The neural network was used as a method of search over the extremely large search space defined by these requirements. This was done by implementing each requirement as a term in the energy function of the Hopfield network.

Neural network techniques rely heavily on an appropriate formulation of the problem and the design of network architecture. At present, there is no structured way to perform either task optimally. It is therefore difficult to produce a solution that can be described as optimal in any way. Whatever formulation is chosen, the resulting
network must be trained to produce a solution both specific enough to give accurate
results and general enough to be robust. Correct training of the network is vital to its
future performance. Unsupervised training schema can lead to entirely unpredictable
segmentation results.

The implicit models created within neural networks seldom lead to a greater overall
understanding of the problem. As we wish to gain a greater understanding of the
effects of applying constraints to our segmentation task, we have chosen not to use
neural networks. Instead, we have chosen to construct explicit models to represent
prior knowledge about capillary shape and appearance.

2.2 Model Driven Methods

Data-driven methods such as thresholding or edge detection rarely give satisfactory
results when used in isolation. Except in the most straightforward cases, image
evidence is often ambiguous and needs to be augmented by some prior knowledge of
the object being segmented. Prior knowledge is often provided by a model of object
characteristics. This section gives a summary of various modelling techniques used
to represent object shape and appearance.

2.2.1 Modelling Shape

In order to use \textit{a-priori} shape information in a non-trivial segmentation task, mod-
elling techniques must represent the shape of an object and the variation found within
the object class. They must also provide a method by which deformations are driven
to fit the image evidence, without invalidating the shape. There are many different
methods available to perform these tasks, and a summary of some research in this
area is given below.
Structural Models

Widrow [149] developed one of the first examples of a whole-boundary shape modelling method. He used parameterised templates called rubber masks to model objects. Rubber masks allowed a certain amount of deformation in the model during fitting. The parameters of the model were the sizes and relationships between subparts. Yuille et al [152] used a parameterised template for an eye, which consisted of two parabolas surrounding a circle. They developed a similar model for the mouth. The models were deformed to the image by optimising a cost function, based on morphological features. Other examples of similar hand crafted models include Hill and Taylor [69], who used a model of the heart together with genetic algorithm search to find the left ventricle in echocardiograms, and Lipson et al [85], who mapped elliptical models of vertebrae onto CT images of the spine. Fritsch et al [53] used models based on medial axes and boundaries with connectedness information. This was used to aid segmentation of the ventricles of the brain and other medical structures.

These models had the advantage of representing objects using a small number of parameters. However, the types of object that can be represented are limited, as in each case enough consistent structure must be present for the models to be generated by hand. This type of modelling does not generalise to represent arbitrary unstructured shapes, such as those found in capillary images.

Physical Models

Pentland and Sclaroff [111] used the metaphor of modelling clay to represent 3D objects. Shape is thought of as the result of pulling, pushing and pinching a lump of elastic material. The mathematical formulation of this is based on the Finite Element Method (FEM) [154], which is a standard engineering technique for simulating the dynamic behaviour of an object. The FEM represents a body as a set of $n$ nodes, a mass matrix $M$ and a stiffness matrix $K$. A disadvantage of this
method of representing objects is that the number of description parameters is often large; both matrices have dimension $2n \times 2n$. This necessitates the use of heuristics such as smoothness and symmetry to reduce the number of parameters. If left unaltered, the high dimensionality of the representation can result in unstable and non-unique descriptions. This drawback was overcome by using a modal analysis of the FEM stiffness parameter matrix. This modal analysis is an eigenmethod (similar to Cootes [36]), in which Free Vibration Modes of the model are produced by solving the generalised eigenproblem as follows:

$$K \Phi = \Omega^2 \Phi M$$  \hspace{1cm} (2.1)

where $\Phi = (\phi_1, \phi_2, \phi_3, \ldots, \phi_{2n})$ is a $2n \times 2n$ matrix of eigenvectors representing the modes and $\Omega^2 = \text{diag}(\omega_1^2, \omega_2^2, \ldots, \omega_{2n}^2)$ a diagonal matrix of eigenvalues associated with each eigenvector.

A useful property of the Free Vibration Modes is that the low order modes correspond to the rigid-body modes of translation and rotation. The next lowest modes are smooth, whole-body deformations that leave the centre of mass and rotation fixed. When there are fewer degrees of freedom in the sensor measurements than in the model, high-frequency modes of vibration cannot be accurate, as there is not enough data to constrain them. They also do not have much effect on the overall shape of the object. Therefore, high-order modes of vibration are discarded so that the number of degrees of freedom of the sensor data is greater than or the same as that of the model.

In order to develop a FEM model from sensor data, a suitable base shape, such as an ellipsoid, must be influenced by data measurements. This is done by attaching virtual springs between the model’s nodes and the data measurements. This implies a correspondence between the model’s nodes and the data points. In most situations, this correspondence is not given and so must be determined automatically. Pentland
and Sclaroff achieved this by projecting data points onto an ellipsoid. They used this technique to develop 3D models of human heads from laser range data [111]. By comparing the parameters of each model, it is possible to distinguish between different heads. They also demonstrated the ability to fuse data from several different viewpoints of an object into a single model that becomes more precise as more views are integrated.

To model the fine variations found in many medical objects, an extension to the FEM modelling technique was proposed by Sclaroff [124]. This counteracted the effect of discarding the high-order physical vibration modes, by using wavelet based displacement maps to store fine detail of a solid 3D model. This technique was used to form detailed 3D models of heads which were used for data compression [127].

One of the disadvantages of this technique is that each object has to be described in terms of deformations from a single prototype object. This imposes an a-priori parameterisation upon the sensor data, and therefore determines the correspondences between the data and the prototype nodes. Sclaroff and Pentland [126][128], proposed a method of automatically determining correspondences between points of different examples of a class of objects. Two objects are modelled separately in the manner described above. The number of modes in the two models is equated by discarding a number of the higher-order eigenvectors of the larger stiffness matrix. Sets of affinities, stored in an affinity matrix, are calculated. These represent the match between each mode - 0 for a perfect match, increasing as the quality of the match decreases. Looking for the minimum entry in each column or row of the affinity matrix can identify corresponding features. This technique was used to find correspondences between silhouetted shapes such as pears, planes and hands.

Martin et al [95] applied these techniques to model the shape of the brain in neuropathology. In this work, deformation due to disease is separated from that due to head shape. This is done by normalising cranial contents using a complete head shape model, then modelling the resulting structures.
Christensen et al [25] used alternative formulations for physically based models. Given a textbook of valid class examples representing the shape and allowable deformations of a given structure, they devised techniques to find the set of transformations that map a subject to this textbook. Two methods are proposed: an elastic-solid model, and a viscous-fluid model. The elastic-solid model is based on the Gibbs form \( \left( \frac{1}{N} \right) e^{-H(u)} \), where \( H(u) \) is the potential energy. Transformations based on this model are subject to restorative forces proportional to the deformed distance. This prevents the textbook being fully deformed to match the subject exactly. To overcome this, the viscous-fluid model was used, in which the restorative forces are allowed to relax over time. These techniques were used to find the deformations that map the textbook onto the subject. As the textbook is already segmented, a segmentation of the subject is gained. This technique was used to segment the various structures of the brain throughout slices of a MR scan.

Pentland’s physical models produce vibrational modes of variation from a single exemplar that are somewhat arbitrary, and are not guaranteed to be representative of the real variation that occurs in a class of shapes. Objects are assumed to consist of materials with known deformation characteristics. Christensen et al modified the model of allowable deformation to produce better results for textbook registration. Methods described later in this section are based on statistical representations of object deformation. For complex shapes such as capillaries, a statistical representation of shape variation reduces the need to manually tailor model deformation characteristics.

### Landmark Based Methods

In order to parameterise a shape using a sub-sampling scheme, correspondences between each shape in a training set must be developed.

Bookstein [11] introduced points called landmarks to represent corresponding positions on a shape. Landmarks correspond to consistently observed features over a
number of examples. Three principal types of landmarks were defined:

1. points marking parts of the object with particular application-dependent significance, such as the centre of an eye;

2. points marking application-independent things, such as curvature extrema;

3. other points that can be interpolated from points of type 1 and 2, such as points marked at equal distances around a boundary between two type 1 landmarks.

Bookstein [10] developed thin-plate splines as a method of interpolating between two sets of landmark points. The basis function of a thin-plate spline is the surface:

\[ z(x, y) = -r \log r \]  (2.2)

where \( r \) is the distance \( \sqrt{x^2 + y^2} \) from the origin. Splines can be thought of as thin metal plates stretching to infinity. Deformations are represented by a bending energy matrix, which can be calculated to warp two sets of landmark points directly onto one another. The rest of the surface is also warped, leading to an interpolated warp for all other points that were not used to fix the bending energy matrix.

The spline maps are decomposed into linear combinations of geometrically independent principal warps. Principal warps are calculated by an eigenanalysis of the bending energy matrix of the calculated thin-plate spline. It is suggested that the principal warp parameterisation can be used to drive conventional multivariate analysis techniques, such as analysis of variance and covariance.

Thin-plate splines were again used by Bookstein [13] to display a guide to interesting regions of shape deformation. Quadratic variation, which is the sum of squared second partial derivatives of displacement, is used to identify areas of significant deformation. The method is designed to highlight areas that appear significant to a human viewing the deformation meshes of the thin-plate spline. Bookstein proposed that
this technique could be used in determining the nature of anatomical abnormalities against an atlas or sample average.

Bookstein [12] also used thin-plate spline deformations to assign point-to-point correspondences between curves of similar but variable shape. A subset of points, called semi-landmarks, are allowed to vary along shape boundaries to minimise deformation bending energy. Modifications of Procrustes shape alignment methods are given to work with them. Landmark-based multivariate analysis can be carried out following these methods. These techniques allow a curving form to be analysed using landmark techniques without requiring traditional anchored landmark points. Recent work by Davies et al [40] uses information theory to optimise landmark correspondences. Both techniques are concerned with removing spurious shape variation that has been introduced by poor correspondences.

Statistical Shape Models

Many modelling techniques are based on statistical analysis of shape parameterisations. Linear analysis of parameterisations of a training set is the most commonly used methodology. There are many possible parameterisations of shape. The most frequently used parameterisations include: Bookstein’s landmarks and related point-based methods, Fourier descriptors and other whole-boundary decompositions. To carry out a meaningful statistical analysis, corresponding variables in the parameterisation must represent corresponding shape features.

Point Based Methods

Cootes et al [35][36] developed a method of shape modelling called Point Distribution Models (PDM). The method is based on landmark points that represent object features. The features must be consistently present throughout the training set. Landmarks are registered to one another using a Procrustes alignment process, which is described in [32]. This process minimises the weighted sum of squared distances be-
between equivalent points on different examples. Residual shape variation after alignment is modelled using a principal components analysis of the aligned covariance matrix. The components are known as *modes of variation* and often correspond to intuitive shape deformations. Insignificant modes can be discarded to produce a more compact representation.

Image search is carried out using 1-D grey-level profile models passing through each landmark position, normal to the boundary. A value of the fit of the profile to the image data is calculated along this normal. From this, the direction in which the profile model best fits the image data can be calculated for each landmark point. The adjustment in the scale, orientation and shape parameters which best fits this desired deformation can then be calculated to give a deformed shape which better fits the image data. This procedure is repeated until the model converges to a stable position. This method is similar to the Active Contour Models of Kass *et al.* [76], but it adds application specific global shape constraints, so the term *Active Shape Model* (ASM) was adopted to describe the method.

The ASM fitting technique is a local refinement method, and as such its performance is dependant on initialisation being close enough to the global solution. Hill *et al.* used global genetic algorithm search to produce a set of initial model positions [69][70]. This method has been applied to segment a wide variety of objects, such as hands, various facial features, resistors and the heart [32][31][142].

PDMs [36] are developed from the statistical variation found in a set of example shapes, and represent the class of interest well, only when a large set of examples is available. Cootes and Taylor [34] developed a technique of combining PDMs with FEM models to aid in the building of an active shape model. When only a few examples of a class of objects are available, the physical modes of vibration of each example influence the model considerably. This is achieved by combining statistical modes and physical modes of vibration of each example, using a method that favours modes that have some statistical evidence for their presence. As the number of
available examples increases, the amount of FEM variation in the PDM covariance matrix decreases, until the model is based only upon the statistical variation observed in the training set. Experimentation showed that this hybrid modelling technique performed better than both FEM and PDM models with a limited number of training examples.

ASM have been successfully used to segment medical images in many studies [32][31][50][79][132][133][142]. As such, they represent a ‘gold standard’ modelling technique and we have applied the technique to the segmentation of capillary images. A full description of the PDM technique is given in chapter 3, along with an evaluation of its segmentation performance.

One of the drawbacks of the Active Shape Models is their reliance on manually placed landmark points. In particular, type 1 and 2 points may not be consistently present in objects to be modelled. They are not present in capillary images.

Another limitation is that PDMs use a linear analysis of covariance. As a result, the models do not represent non-linear shape variation well. A common example of non-linear shape variation is bending, which requires landmark points to follow curves as each shape parameter is varied. Deformations like this are modelled linearly by combinations of modes of variation, which display nonlinear relationships.

Several solutions to this problem have been proposed. Sozou et al [135] created shape modes that allowed landmark points to vary along polynomial paths using polynomial regression. Hinshaw et al [71] proposed a radial contour model, which consisted of a number of equally spaced boundary points represented as polar coordinates normalised with respect to a principal axis. The covariance matrix of these coordinates over the training set is calculated to add the a-priori shape information to the model. This model was used within an interactive segmentation system, in which the user was able to make adjustments to a single radial contour to improve segmentation. The transformation of landmark coordinates to a radial basis linearises bending relationships. This can be advantageous if significant bending is observed in
the training data. A similar technique was used by Heap and Hogg [64], this time using a combination of cartesian and polar coordinates to linearise bending effects.

Bending is not the only source of non-linear variation. In general, training examples can have an arbitrary distribution in model parameter space. Bregler and Omohundro [16] proposed representing the constraint surface with a union of lower dimensional hyperplanes in parameter space. Heap and Hogg [65][66] extended this idea to represent shape constraints piecewise, with a union of hyperellipsoid bounded regions. Ellipsoids represent overlapping clusters of training data in model parameter space, which are formed by $k$-means clustering. Shapes can be constrained by projecting their parameters towards the nearest hyperellipsoid.

Techniques also exist to represent arbitrary non-linear distributions without using a piecewise linear approximation. Romdhani et al [118] applied a non-linear Kernel PCA [123] to face shape data. However, Twinning and Taylor [145] later showed that an alternative ‘proximity to data’ function was required to constrain shape correctly. Non-linear shape modelling using Kernel PCA is an active research area.

**Whole Boundary Methods**

Many whole-boundary parameterisation schemes are available. Some of the most commonly applied methods are based on a Fourier decomposition of boundaries. Many shapes, including capillary boundaries, can be represented by closed boundaries. A closed curve can be represented with a periodic function $z(t)$, where $t$ is the distance around the boundary. If the total boundary length is defined to be $2\pi$, then a Fourier analysis can be used to decompose $z(t)$ into its constituent frequencies:

$$z(t) = \sum_n T_n \exp^{-i\omega t}$$

(2.3)

The series coefficients $T_n$ are called the *Fourier descriptors* of the original signal $z(t)$. In practice, this summation must be truncated. Most boundaries of structures in nat-
ural images are band–limited in the frequency domain, so truncating the summation need not lead to a poor representation of boundary shape.

Many Fourier parameterisations of boundaries are presented in the literature [60][109][112][136][153]. Fourier descriptors represent a promising method of capillary boundary parameterisation and have been compared to PDMs. A full description of Fourier descriptors is given in chapter 3, together with an evaluation of their properties.

### 2.2.2 Modelling Appearance

Techniques have been developed to model the grey–level appearance and variation of objects. Prior information about the appearance of objects can be used to increase segmentation performance.

Turk and Pentland [144] were amongst the first to formulate a method of grey–level description. They use Principal Components Analysis (PCA) to describe the grey–level appearance of face images in terms of a set of basis functions, or *eigenfaces*. This representation is not robust to shape changes, and does not deal well with changes in overall lighting and facial expression, as there is no normalisation for these factors. However, an advantage of the method is that models can be fitted to images using standard, fast, correlation-based methods. Eigenface model matching techniques were improved by Moghaddam and Pentland [100] who used multivariate Gaussian and Mixture–of–Gaussians functions to model membership probabilities for automatic recognition. They demonstrated their technique by finding and normalising faces and hands in images.

Nastar and Pentland [103] modelled grey–level appearance as a 3D intensity surface. Surfaces are modelled in a physically based manner, with associated mass, damping and stiffness matrices. To determine the similarity of two images, the surface of one is allowed to deform dynamically based on the external force applied by the other image. A strain matrix is formed from this calculation. The low-order strain energies
in this matrix are used as a measure of similarity. A recognition rate of 97% was achieved over a sample of 200 people from a database of 7,562 images.

Edwards et al [48] extended the Active Shape Models of Cootes et al [36] to include grey-level and shape information. The model consists of a PDM of landmark points marked on example images, which in this case were faces. Each face is warped to the mean shape of the PDM, using a triangulation method, and then normalised against the effects of global lighting changes. The grey-level information within the normalised faces is then decomposed using PCA producing appearance parameters for each training face. A further PCA is carried out to represent any correlations between shape and grey-level variation. This results in a set of appearance parameters that control both shape and grey-level appearance. This technique was used to produce photo-realistic models of faces, and could be used to approximate unseen data that had been marked up with the appropriate landmarks. However, objects represented in this way require a high number of appearance parameters (faces are typically represented by between 50-100 parameters). This high dimensionality of the search space makes conventional search techniques impractical.

To overcome this drawback, Cootes et al [30] developed a method of optimisation tailored to the combined shape and appearance models. The two techniques are known collectively as Active Appearance Models (AAM). Following the work of Covell [37] and Black and Yacoob [9], image difference patterns corresponding to changes in each model parameter are learnt and used to modify the model estimate. This association is assumed to be linear and is learned off-line during the training of the AAM using a multi-variate regression of known parameter offsets.

Edwards et al [49] used AAMs to extract identity and expression appearance parameters. The models were used to recognise and track individuals and their expressions.

A similar technique called Active Blobs was developed by Sclaroff and Isidoro [125]. Here, search is driven by learned correspondences between model fit error and parameter offset in a similar way as above. However, in this case, models are derived
from a single example, with variation represented by low energy mesh deformations, calculated using a Finite Element method analysis. Cootes et al [30] suggest that this technique may be useful in bootstrapping an AAM in a similar way to that suggested for ASMs in [33].

Nastar et al [102] developed appearance models that combine both physical and statistical modes of variation. Their parameterisation built on earlier work in which images are represented by 3D intensity surfaces [103]. The technique, however, requires images to be aligned by some preprocessing and does not contain an explicit representation of the structure of objects. Because of this, the marking up of images with landmark points is not necessary.

## 2.3 Model Fitting

Segmenting images with models requires the determination of model parameters to fit the model to underlying image data in some optimal manner. The optimal fit of the model to the data should correspond to the desired segmentation of the image. Usually, a model fit function is defined. This function should encapsulate the likelihood of the underlying image evidence, given the model. For example, ASM fitting uses searches of local image regions for the best match to a model of appearance [36]. The best fit position is defined to be the point at which the Mahalanobis distance between the model and the data is minimised. In this case, the Mahalanobis distance gives a measure of the closeness of fit between the profile model and the image data.

Model fitting can be thought of as maximising (or minimising) fit function value over the space defined by model parameters. This section gives an overview of commonly used methods to find maximum function values. We have only considered algorithms that are directly applicable to fitting models to capillary images.

Firstly we give details of local refinement methods of model fitting. Global optimi-
sation techniques are considered in section 2.3.2. Finally, we consider the need for error tolerant model parameter estimation as part of the fitting process.

Many of the modelling techniques we have outlined in section 2.2 have their own associated fitting algorithm.

### 2.3.1 Local Refinement

Local refinement methods aim to move model parameters in the direction of the gradient of the fit function. The technique is known as gradient ascent, or hill climbing. Fit function gradient can not usually be calculated analytically as it depends on unknown image data. A local refinement of the position of a model in an image which results in a better fit to local image data is usually used to estimate model parameter changes. Commonly, image data is used to deform the model instance towards a better solution. This deformed data is then transformed back into model parameters. Examples of this technique that have already been mentioned are ASM fitting [36], snakes [76].

Active appearance models [46] use a linear model of the relationship between image errors and parameter changes to achieve local refinement. The model is learned from known parameter offsets during model building.

A larger image area can be search efficiently using multi-resolution techniques. for example, multi-resolution ASM model fitting uses profile models built from each level of a sub-sampled Gaussian smoothed image pyramid. This involves searching the image at a coarse resolution, and then refining the location in a series of finer resolution images. Because pyramid images are sub-sampled, pixels at coarse resolutions represent a larger image area than those at finer resolution levels. This allows a larger image area to be considered during model fitting. The multi-resolution ASM fitting algorithm is described in full in section 3.2.2.
Gradient ascent search may become trapped in local maxima. Point (a) will not find the global maximum of the fitness function. Point (b) has a better initialisation and will find the true maximum.

The performance of local refinement model fitting algorithms is dependent on the characteristics of the fit function search space and the initialisation of the search. If the search space contains more than one locally optimum value, a local refinement process will only find the nearest. If a good initialisation is not available, this optimum is not guaranteed to correspond to the global optimum. This is illustrated in figure 2.1. Multi-resolution gradient ascent techniques can be used to reduce the effects of this problem, whilst increasing search efficiency [28].

2.3.2 Global Optimisation

When image information is complex and ambiguous, it is likely that the model parameter space will contain many local maxima. The search space defined by model parameters typically has a large number of dimensions, making exhaustive search impractical. An efficient search method is required to find the global optimum fit function value.
One of the most commonly used global optimisation strategies are genetic algorithms [41]. Genetic algorithms use a function maximisation approach analogous to natural selection. Candidate solutions are encoded as *chromosomes*, which usually consist of vectors of binary or floating point numbers. A population of solutions is maintained. Optimal solutions are evolved from this population by combining members of the current population to form a new population. Each chromosome has an associated *fitness value* that determines the likelihood of it propagating to the next generation. New individuals (children) are created (bred) using combinations of existing solutions (parents). Several operators analogous to genetic rearrangement have been suggested to apply during combination. The most common are *crossover* and *mutation*. In crossover, the chromosomes are split at a random location and opposing sections are rejoined to form two new chromosomes:

\[
(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8) \rightarrow (a_1, a_2, a_3, b_4, b_5, b_6, b_7, b_8)
\]

\[
(b_1, b_2, b_3, a_4, a_5, b_6, b_7, b_8) \rightarrow (b_1, b_2, b_3, a_4, a_5, a_6, a_7, a_8)
\]  

(2.4)

Floating point crossover selectively ‘bleeds’ values into one another

\[
(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8) \rightarrow (a'_1, a'_2, a'_3, b'_4, b'_5, b'_6, b'_7, b'_8)
\]

\[
(b_1, b_2, b_3, a_4, a_5, b_6, b_7, b_8) \rightarrow (b'_1, b'_2, b'_3, a'_4, a'_5, a'_6, a'_7, a'_8)
\]  

(2.5)

where

\[
a'_i = ra_i + (1 - r)b_i, \quad b'_i = rb_i + (1 - r)a_i
\]  

(2.6)

and \( r \) is a random number in the range \([0 \ldots 1]\).

Binary mutation randomly selects a small number of chromosome values to invert in each population. Floating point mutation can change values randomly by a small amount, or set a value to one end of its valid range. Random values can be taken from
uniform or normal distributions. The width of the Gaussian random distributions are typically reduced as the maximum number of generations approaches. This has the effect of focussing the population on a particular parameter range as evolution proceeds.

The assumption is made that the optima can be formed by combining small subsections of the chromosomes, each of which will tend to improve the function evaluation of any full chromosome containing them. Thus these good building blocks will tend to be reproduced and will propagate through the population. Eventually all the current generation of individuals should converge to the global optimum.

Genetic algorithms have previously been used to fit PDMs to image data, both with \cite{69} \cite{70} and without \cite{68} an embedded local ASM refinement step. Hill et al \cite{69} showed that the efficiency of GA convergence can be improved by fitting each individual to the local image data at each generation.

2.3.3 Robust Fitting

In most cases, search is directly driven by image information. If image data is extremely complex or ambiguous, candidate model positions will also exhibit these characteristics. In all but the most ideal of cases, information derived from image data will result in a model realisation that is not a valid example of the object class. Model parameterisations are used to constrain the interpretation back to the region of valid class examples.

Sometimes, data is so inconsistent that simply using model parameter constraints does not result in a good fit of model to data. A more robust method of model parameter estimation must be used. Typically, model parameter estimation is optimal with respect to certain criteria. In the case of PDMs, a least squares estimation of model parameters has been used. Least squares fitting is sensitive to noisy data with outliers. This is demonstrated in figure 2.2, where a model of a line is fitted to point
Chapter 2. Image Interpretation Methods

Figure 2.2: Fitting a line model to data using a least squares method is sensitive to outliers. The dotted line results from fitting a line to the data points marked as blue crosses. The solid line represents the ideal model fit for this data.

data. The model is excessively moved from its ideal position by the single gross outlier. If the data used for parameter estimation contains a significant proportion of outliers, a least squares algorithm will not result in a good estimation of model parameters. Error tolerant model fitting is required.

Several generally applicable approaches to the problem exist [80][98][97]. All offer robust model parameter estimation by either utilising different model fit measures, such as least median of squares, or using a subset of data to estimate parameters. The method we have chosen to utilise is Random Sample Consensus (RANSAC) model fitting [51]. RANSAC allows a definable number of outliers to be ignored during parameter estimation. A randomly chosen subset of data is used to estimate model parameters. If these parameters are consistent with enough original data, the parameterisation is accepted. Otherwise, the process is repeated. We have used RANSAC parameter estimation to robustly estimate PDM parameters. A full description of
the technique, and our evaluation is given in section 3.3.3.

2.4 Summary

Many of the data driven techniques mentioned in section 2.1 have been applied to the segmentation of nerve capillaries in a previous study by Byrne [18]. Results from Byrne’s work indicated that data–driven segmentation was insufficiently robust to be applied in a research setting.

Byrne’s study utilised texture information. In chapter 5, we extend his investigation of the textured appearance of nerve capillary structures.

Segmentation robustness can be improved by the use of constraints from prior knowledge. Specific prior knowledge about object shape and appearance is most commonly represented as a model. Models must represent information about the object and how it can vary. Statistical models built from training examples are an ideal solution. These models offer much more flexibility than structural models. Modelling observed variation is more appropriate for highly variable capillaries than the rather arbitrary assumptions about allowable elastic deformation imposed by FEM techniques.

Statistical models can only be built once a consistent parameterisation of shape and/or appearance has been made. PDMs use a landmark representation of shape, and have been successfully applied many times to medical image analysis. We have chosen to evaluate their performance as a ‘gold standard’ parameterisation. Capillary shape does not lend itself to representation by landmarks as there are no obvious consistently observable features. A whole–boundary parameterisation may be more appropriate. In chapter 3 we evaluate Staib and Duncan’s elliptic Fourier descriptors [136] as a method of boundary parameterisation, and compare them to PDMs. Appearance models offer a method of representing appearance information as a further source of prior knowledge. In section 3.6, we evaluate the characteristics of Edwards
et al appearance models when applied to capillary images.

None of the shape modelling techniques that we have described are capable of representing features, such as the LEC boundary, which are only observable in certain capillary images. We address this problem in chapter 4, where a novel extension to PDMs is described.
Chapter 3

Modelling Shape and Appearance

3.1 Introduction

This chapter describes three modelling techniques that we have applied to nerve capillary structures. Nerve capillaries exhibit a large range of shape and appearance variation. A good model of capillaries will be general enough to encapsulate all observed variability, and specific enough to represent only valid examples of the object. For any complex object, one practical way of establishing allowable variation is by learning from a set of examples.

Two methods of representing capillary shape have been evaluated. Active Shape Models (ASMs), introduced by Cootes et al [36], parameterise shapes with consistently placed landmark points. Staib and Duncan’s Fourier Descriptors [136], parameterise closed boundaries with a continuous representation. As there are no obvious physically consistent features around the boundaries between different examples of capillary structure, a whole–boundary representation appears to be a more appropriate modelling paradigm. However we show that the characteristics of the Fourier parameterisation do not produce a compact or general model of capillary shape.
The third technique we describe is Active Appearance Models (AAMs) [47] [30]. AAMs include a method of representing the grey-level appearance of nerve capillaries. However, the highly variable shape of the nerve capillaries requires some modifications to the appearance modelling algorithm. This variability severely restricts the usefulness of using AAMs in representing nerve capillary objects.

The suitability of each modelling technique has been evaluated. In section 3.3, evaluations of modifications and enhancements to the standard ASM implementation are presented. Sections 3.5 and 3.7 describe evaluations of the limitations of Fourier descriptors and appearance models when applied to capillary data.

### 3.2 Active Shape Models (ASMs)

ASMs are a statistical shape model together with a model fitting algorithm. The shape model is formed from a linear statistical analysis of a training set. The training data consists of lists of consistently placed landmark points, and the resulting model is known as a point distribution model (PDM) [29]. The method has an associated local refinement algorithm to fit PDMs to image data whilst restricting shape to a statistically defined valid range. ASMs have been used successfully in many medical image analysis applications [32][31][50][79][132][133].

#### 3.2.1 Modelling Shape

A PDM is constructed in three steps: first landmarking the training set, then aligning the landmarks, and finally analysing the remaining variation amongst the aligned training data. Each of these steps is described in the following sections.
Landmarking the Training Set

To model an object we must first parameterise its shape. PDMs achieve this with lists of landmark points, placed in physically consistent positions. The term was introduced by Bookstein [11] in which he defines three principal types of landmarks:

1. points marking parts of the object with particular application-dependent significance, such as the centre of an eye;
2. points marking application-independent things, such as curvature extrema;
3. other points that can be interpolated from points of type 1 and 2, such as points marked at equal distances around a boundary between two type 1 landmarks.

In this way, a particular landmark represents the position of the same physical feature across the images in the training set. Structures in training example $i$ are represented by a list of landmark coordinates as follows:

$$x_i = (x_1, x_2, x_3, \ldots, x_n, y_1, y_2, y_3, \ldots, y_n)^T$$

where $(x_j, y_j)$ are the coordinates of landmark $j$ and the structures are represented by $n$ landmark points.

Aligning the Landmarks

We require a PDM which encapsulates all object specific variation but not variation due to the object’s relative position, scale or orientation in the image frame. For this reason the landmark points of the training set are aligned to one another. The generalised Procrustes analysis method [59] is used. This aligns each shape so the sum of Euclidian distances of each training shape to the mean shape is minimised in a weighed least-squares sense. Weights can be chosen to give more significance to
the points that tend to be most ‘stable’ over the training set. Further details of the alignment procedure are given by Cootes et al [36].

**Modelling Shape Variation**

The Procrustes analysis results in a set of aligned training shape vectors, \( x_i' \) with dimensionality \( 2n \). The number of degrees of freedom with which the shapes can vary is typically much less than \( 2n \). This is because the variation in landmark position between examples is usually highly correlated. PDMs use Principal Component Analysis [92] to capture these correlations and therefore reduce the number parameters required to represent the shape. The approach is as follows:

The mean shape, \( \bar{x} \), is given by:

\[
\bar{x} = \frac{1}{s} \sum_{i=1}^{s} x_i'
\]  

(3.2)

where \( s \) is the number of training examples.

The \( 2n \times 2n \) covariance matrix, \( S \), of the data is:

\[
S = \frac{1}{s-1} \sum_{i=1}^{s} (x_i' - \bar{x})(x_i' - \bar{x})^T
\]  

(3.3)

The training data, \( x \), forms a cloud of points in a \( 2n \)-D space. The eigenvectors, \( p_j \), and corresponding eigenvalues, \( \lambda_j \) (\( j = 1, \ldots, 2n \)), of \( S \) represent a set of orthogonal axes which are aligned with the principal *modes of variation* of the cloud. The eigenvectors corresponding to the largest eigenvalues represent the most significant modes along which the shape can vary. \( \lambda_j \) gives the variance along the \( j \)th component. Most of the shape variation can be represented by selecting a smaller number, \( n_s < 2n \), of these axes which explain a large proportion of variation. Often \( n_s \) is chosen so the selected axes explain at least, say 95\%, of the variance exhibited in the training
Neglecting any alignment steps, any shape, $x_i$, in the training set can then be approximated by a weighted sum of the first $n_s$ eigenvectors and the mean shape:

$$x \approx \bar{x} + P b_s$$

(3.4)

where $P = (p_1, p_2, \ldots, p_{n_s})$ is the matrix of the first $n_s$ eigenvectors, and $b_s$ is a $n_s$ dimensional vector of weights, normally referred to as shape parameters.

The shape parameters, $b_s$, which best match the model to a particular shape vector, $x$, can be calculated as follows:

$$b_s = P^T (x_i - \bar{x})$$

(3.5)

$b_s$ defines a set of $n_s$ model parameters.

PDMs are generative models. New examples can be constructed by varying the values of the shape parameters, $b_s$, in equation 3.4. It is important to define limits on the range of values that the shape parameters can take. Assuming an independent uni-modal normal distribution for each parameter, $b_k$, the limits are chosen to be:

$$-r\sigma_k \leq b_k \leq r\sigma_k$$

(3.6)

where $\sigma_k$ is the standard deviation of shape parameter $k$ ($\sigma_k = \sqrt{\lambda_k}$). $r$ is chosen such that any shapes generated are plausible instances of the class of objects to be represented.

Figure 3.1 shows the effect of varying the three most significant modes in a model of nerve capillary shape. The basement membrane/endothelial cell material (BMEC) and lumen/endothelial cell material (LEC) boundaries have been included in the
Figure 3.1: Varying the first three modes from a nerve capillary PDM between $\pm 2\sigma_k$ about the model mean.
model. The modes have been varied by $\pm 2\sigma_k$ from the model mean which, assuming a Gaussian distribution of model parameters, encompasses 95% of all valid examples of nerve capillary shape. The landmarking scheme used to annotate the capillary structures in the training set is described in section 3.3.1.

### 3.2.2 Model Fitting

The ASM model fitting algorithm combines local models of grey-level appearance with the statistical \textit{a-priori} shape constraints of the PDM in an iterative refinement scheme. In this way, the algorithm fits the model to the image data whilst retaining the overall validity of the shape of the final fit. The technique is similar to the ‘snakes’ of Kass \textit{et al} [76] in that image data is used to ‘attract’ control points; however, the application of global \textit{a-priori} shape constraints is of significant benefit in many complex applications, hence ASMs are sometimes known as ‘smart snakes’.

#### Grey-Level Profile Models

To model the local grey-level appearance in the area around each landmark point, PCA models are built [32]. Each model usually represents a region aligned to the normal to the curve on which the landmark point lies, i.e. it represents a profile across the shape boundary at the landmark position.

We can calculate the similarity between a local model and any image patch of the same dimensions using a statistically defined ‘fit’ function. The quality of a model’s fit to a particular image patch is inversely proportional to the Mahalanobis distance of the image data from the model distribution. The Mahalanobis distance $m_{ij}$ for a landmark model at position $(i,j)$ is calculated as:

$$m_{ij} = (x_{ij} - \bar{x})^T S^{-1} (x_{ij} - \bar{x})$$  \hspace{1cm} (3.7)
where $\mathbf{x}_{ij}$ is the data vector sampled from the image region at position $(i, j)$, $\mathbf{\bar{x}}$ is the mean of the profile model and $\mathbf{S}$ is the covariance of the profile model.

**Iterative Image Search**

To begin ASM search, an instance of a PDM is initialised in an image. The position of the PDM is repeatedly refined to improve the fit of the local grey-level models to the image data. This is achieved by searching the image data around the current location of each landmark point, seeking the position of the best local fit of each model. Usually the search is carried out along normals to the curve on which the point lies.

This results in a set of suggested adjustments to the landmark positions, given by a displacement vector, $d\mathbf{x}$:

$$
    d\mathbf{x} = (dx_1, dx_2, \ldots, dx_n, dy_1, dy_2, \ldots, dy_n) \quad (3.8)
$$

Given the vector of required adjustments, $d\mathbf{x}$, the model is updated in two stages:

1. the pose, scale, and orientation of the model are updated.
2. the shape parameters, $b$ are updated.

The pose, scale and orientation of the model are adjusted to align the current instance of the model to the new landmark positions in a least-squares sense. The remaining differences between the model points and their desired locations are known as the residual displacements, $d\mathbf{x}'$. The shape parameters of the model are now updated to minimise these displacements whilst retaining the shape constraints imposed by the PDM. This is done by allowing the model parameters, $b$, to vary only within limits learnt from the training set. Cootes et al [36] show that by applying a least squares approach, the optimum adjustments, $db$, to the shape parameters, $b$, are given by:
By ensuring that the model points are only moved by changing the model parameters within limits learnt during training, the new shape will always represent a legal example of the class of shapes represented by the model. The search procedure is repeated a fixed number of times, or until further iterations do not result in any significant change in the model. At this point, the search is said to have converged.

**Multi–Resolution Image Search**

A multi–resolution extension to standard ASM search algorithm improves the efficiency of searching large image regions [28]. Images are searched at a coarse resolution, and results are repeatedly refined using higher resolution image data.

For each image in the training set, a Gaussian image pyramid is built [17]. The base image (level 0) is the original image. The next image (level 1) is formed by smoothing the original image, then sub-sampling to obtain an image with half the number of pixels in each dimension. Subsequent levels are formed using the same process.

Separate profile models for each level of the pyramid are constructed. Usually, the size of the local image patches remains constant at each image resolution. Since the pixels at level $L$ of the pyramid represent an image region $2^L$ times the size of the original pixels, the coarser profile models represent a larger image area. We need only search a local region of a few pixels around each landmark point. At coarse image levels, this allows quite large movements.

Resolution levels are switched when a proportion $P_{\text{converge}}$ (typically around 90%) of profile model best fit matches are found to lie within the 50% of the profile model region. When the proportion of points meeting this criteria exceeds $P_{\text{converge}}$, the search is said to have converged at the current resolution.
The entire algorithm can be represented as:

1. Set $L = L_{max}$.

2. While $L \leq 0$
   
   (a) Compute model point positions in image at level $L$.
   
   (b) Search local image region at each model point.
   
   (c) Update model parameters to profile model best fit positions.
   
   (d) Return to 2(a), unless more than $P_{converge}$ of the points are found to be close to the current position, or $N_{max}$ iterations have been applied at this resolution.

   (e) If $L > 0$ then set $L = L - 1$.

   (f) Increase model scale by factor of 2.

3. Final result is given by the parameters after convergence at level 0

The algorithm can be tuned to a particular task by setting the values of three parameters:

1. $L_{max}$: the number of resolution levels in the image pyramid,

2. $N_{max}$: the maximum number of iterations at each resolution level,

3. $P_{converge}$: the proportion of profile model fit positions required to determine convergence.

### 3.3 Application of ASMs to Capillary Images

The standard ASM algorithm has been optimised for use with nerve capillary objects. In section 3.3.1, a suitable landmarking scheme has been developed to parameterise
the BMEC and LEC boundaries. There are many parameters involved in the con-
struction of ASM grey-level profile models. Section 3.3.2 describes evaluations that
have been carried out to optimise each variable. Several extensions to the standard
profile modelling algorithm have been evaluated. They include an explicit representa-
tion of locally misleading image evidence and an alternative, error tolerant RANSAC
model fitting algorithm, described in section 3.3.3. The results of each evaluation are
presented in section 3.3.5.

3.3.1 Landmarking Strategy

PDM landmarks typically represent physical features that are consistently visible
across the training set. However, the BMEC and LEC boundaries do not appear to
contain any obvious consistent features. The boundaries appear roughly elliptical in
shape with seemingly random and frequently large deviations from this simple model.
As well as physically consistent landmarks, PDM landmark lists usually contain what
Bookstein described as type 3 landmarks, i.e. points evenly spaced along a structure
between other landmarks. These are typically required to refine the parameterisation
of the shape of object boundaries.

Bearing these points in mind, the following method of landmarking the nerve capillary
structures has been devised. The scheme is illustrated in figure 3.2. Firstly, we
have defined one physically consistent landmark position which lies at one end of
the major axis of the best fitting ellipse to the BMEC boundary. This intersection
is shown in figure 3.2(b) and labelled point 1 in figure 3.2(c). Starting from this
point, we place further landmark points around the BMEC boundary evenly spaced
anticlockwise in Euclidian distance. The landmarks are therefore aligned to the gross
orientation of the capillary. Aligning the boundary parameterisation in this way
removes information about the orientation of the capillary in the image. We do not
have any reason to believe that the alignment of the capillaries in the images has
any physical significance, so this is a reasonable step to take. The positions of each
evenly spaced landmark point are calculated using spline interpolation of the original annotation boundary. It is not practical to use points evenly spaced in $\theta$, as the curves are often re-entrant. This leads to multiple intersects for some radial spokes. Figure 3.2(c) shows the resulting landmark points for the BMEC boundary. For simplicity of display we have only shown 20 points on the BMEC boundary, however our final landmark lists contain 50 evenly spaced points representing the BMEC boundary.

A second consistent landmark position is defined as the basis for the representation of the LEC boundary. The major axis of the best fitting ellipse to the BMEC boundary is offset to lie through the LEC boundary’s centre of gravity. A landmark is placed at the intersect of this axis with the LEC boundary. Of the two possible intersection points, the one that is closest to the first landmark point on the BMEC boundary is used as the first landmark in the LEC boundary parameterisation. This ensures the consistent relative orientation of both boundaries. Landmarks are evenly spaced around the LEC boundary using the same method as the BMEC boundary parameterisation.

An example of a complete landmarking is shown in figure 3.2 (d). Only 20 points are shown in the figure, whereas the representation we have used in our evaluations contains 50 points to represent the LEC boundary. The landmarks of the LEC boundary are aligned to the major axis of the BMEC boundary. This removes relative orientation differences between the two sets of boundary landmarks that would occur if we aligned each boundary by its own major axis. As there appears to be no consistent relationship between the gross orientation of the BMEC and LEC boundaries, internal orientation differences would be unrelated. This would lead to a unreliable alignment of shapes and therefore data unsuitable for modelling with the linear statistical analysis.

There is considerable fine detail in each annotated capillary boundary. It is not realistic to be able to represent this detail using only 50 landmarks and a training set of 131 boundaries. For this reason, annotated boundaries are smoothed before landmarks are generated. The smoothing process uses the property of Fourier descriptors,
whereby truncating the parameter vector removes high frequency content, whilst still guaranteeing a closed boundary [87]. Fourier descriptor smoothing is described in section 3.4. We have chosen to smooth each boundary using 10 harmonic ellipses. This amount of smoothing has been observed to give good tradeoff between removing fine detail whilst retaining all large scale boundary shape.

Each capillary in the training set is represented by 100 landmark points. This gives a data vector with 200 elements (equation 3.1) for each training example. When we wish to construct models representing only the BMEC boundary, we use the first 50 landmarks from each example. The annotations that we have available are described in section 1.4.2 (page 31).

In many capillary images, the lumen has become so constricted as to be practically unobservable. In these cases the landmarks representing the LEC boundary are not available. Standard PDMs cannot be built with incomplete data. Models capable of representing this data are described in chapter 4. In this chapter, when a PDM is built that contains both BMEC and LEC boundaries, we have used the subset of training examples that contain complete annotations of both boundaries, discarding any incomplete annotations.

### 3.3.2 Profile Model Optimisation

This section describes work to optimise grey-level landmark profile models for ASM image fitting. The models represent the local appearance of probabilistic texture images, generated using biorthogonal wavelet texture features. The algorithm used to produce the texture images is described in chapter 5.

Several parameters can be varied when building ASM profile models. Changing each variable results in changes in model fitting performance. To find the best ASM fitting performance for texture processed images, we have evaluated the effects of the following factors:
Figure 3.2: Landmarking nerve capillary structures: (a) shows the original annotated BMEC and LEC boundaries, (b) shows the best fitting ellipse to the BMEC boundary together with its major axis. (c) shows the 20 evenly spaced landmark points representing the BMEC boundary starting at the intersection of the BMEC with the major axis of the ellipse. (d) adds the 20 evenly spaced landmark points representing the LEC boundary, starting at the intersect with the major axis, which has been offset to lie through the LEC boundary’s centre of gravity.
• data normalisation,
• combined grey-level statistics,
• data selection,
• explicit multi-class mixture models,
• profile model patch size,
• total variance.

In the following section, we give a detailed description of each, and the rational for its investigation.

In all cases, the effect of the factor is evaluated by leave-one-out cross-validation. Results are shown collectively in section 3.3.5, together with results from evaluations of other model fitting strategies.

**Normalisation**

The grey-level appearance around the BMEC and LEC boundaries is extremely varied across the training set. Even the most carefully designed texture measure will produce significantly differing responses from image to image. Normalising the image evidence used during profile model building and subsequent ASM model fitting, should improve the overall performance of the system. To evaluate this, we have performed a leave-one-out model fitting experiment as follows. A subset of 10 images was chosen at random from the training set. These images were searched with an ASM constructed from the entire training set after removing any annotations associated with the search image. The ASMs were built with BMEC boundary landmarks. In total, 131 annotations were available to train the PDM and profile models.

ASM search is a local refinement process, therefore model fitting performance is dependent on initialisation conditions. The texture images are extremely complex.
Changes in model fitting performance that we wish to observe may be obscured by the complexity of the image evidence if poor initialisations are used. For this reason, we have chosen to evaluate model fitting performance given the best possible initialisation. The best fit of the model to an image annotation was calculated (equation 3.5) for each search. Model fits were initialised from the resulting shape and pose parameters. In this way, model fitting performance has been evaluated given the best possible starting position in each image.

Each search consisted of five iterations of the ASM fitting algorithm with a search range of 20 pixels in the direction of the normal to the candidate boundary and 3 pixels along the tangent.

Two leave-one-out evaluations were performed. The first used a standard profile model built from un-normalised image data. In the second, profile data values were normalised. Normalisation consisted of scaling data values of each profile so that they sum to 1. This normalisation was carried out both during model building and model fitting. Statistics for initialisation positions and model fitting results are given in section 3.3.5, table 3.1.

**Combined Model**

The BMEC boundary is represented by 50 landmarks, each with its own model of local image appearance. The size of the image patches around each landmark can be specified. In our case, choosing patch dimensions of $5 \times 40$ is not unreasonable. This gives us a vector of 200 elements for each landmark in each image. With around 50 training examples to build each model, true model parameters cannot be estimated accurately.

There appear to be no obvious consistent physical features on either the BMEC or LEC boundaries. To increase the robustness of profile model statistics, data from each landmark around the BMEC boundary has been combined into a single model.
This has been done because we expect no structured variation in profile appearance between any two points on the boundary. It may be true that there is some consistent underlying grey-level structure. However, expert observers are unaware of such structure, and we have defined the landmark positions on the basis of a simple geometric criterion. We therefore take the variation in appearance around the landmarks as being statistical in nature, and treat them as instances of the same model.

Building a combined model increases the number of training examples available for each profile model by a factor of 50, significantly increasing the robustness of the analysis. A combined profile model can be built in the same way for the landmarks of the LEC boundary, which also displays no consistent structural variation.

Figures 3.3 and 3.4 show the effects of varying the first mode of variation of a profile model between $\pm 2\sigma$ from the mean of the model. Figure 3.3 shows the first mode of a typical profile model associated with a landmark on the BMEC boundary. Figure 3.4 shows the first mode of the BMEC boundary combined profile model.

The combined model represents a more consistent change from low values of texture (outside BMEC boundary) to high values (inside the boundary) than the individual landmark model, which contains a greater amount of detail and structure. This structure is associated with the specific data at that landmark point, whereas the smoother combined model represents the overall variation of an arbitrary profile across the BMEC boundary.

If the statistics of a combined profile model are more representative of profile appearance, this improvement will be translated into an improvement in model fitting results. A decrease in image search performance indicates that useful information about the structure of appearance around the boundary has been discarded. A leave-one-out model fitting experiment has been performed to evaluate the effect of using combined profile models. Results, comparing normalised ASM model fitting with the normalised combined model ASM search are shown in section 3.3.5, table 3.2.
Chapter 3. Modelling Shape and Appearance

Figure 3.3: Varying the first mode of a typical grey-level profile model associated with a landmark point on the BMEC boundary between $\pm 2\sigma$ of the mode mean.

Figure 3.4: Varying the first mode of the combined grey-level profile model for the BMEC boundary between $\pm 2\sigma$ of the mode mean.
Profile Data Selection

The quality of profile model statistics can be improved further by manually selecting training data. ASM searches are carried out on images generated by texture analysis (see chapter 5). The analysis has been designed to maximise discrimination between regions on either side of the BMEC boundary. Therefore, the ideal appearance of a profile across this boundary will contain a step intensity edge. Figure 3.5(a) shows an example texture image with a consistent intensity edge at the BMEC boundary. Capillary images are highly variable and complex and there is a great deal of ambiguity in the precise positioning of the various structural boundaries. For these reasons, an obvious intensity gradient is not always present at each landmark. If too many unrepresentative examples are present in the profile model training set, the resulting model will not be specific enough to result in good profile match positions.

A profile model training data selection criterion has been designed as follows:

\[ \bar{x}_{\text{inner}} > \bar{x}_{\text{outer}} + t \]

where \( \bar{x}_{\text{inner}} \) is the average intensity value on the inside of the BMEC boundary, and \( \bar{x}_{\text{outer}} \) is the average value on the outside of the boundary. \( t \) is a threshold value that controls the magnitude of the step edge. We have used a value of \( t = 0.75n_s \), where \( n_s \) is the number of image samples in each profile vector. This value was chosen by observation. Figure 3.5(c) shows example profiles selected in this manner.

Results obtained from leave-one-out ASM searches using selected data are presented in section 3.3.5, table 3.3.
Chapter 3. Modelling Shape and Appearance

Figure 3.5: Example texture images and associated profile data. In each image the BMEC boundary is marked as a yellow line. (a) shows an image with a good intensity edge at the BMEC boundary. (c) shows example profiles across the BMEC boundary from this image. (b) contains many confusing regions close to the BMEC boundary, and (d) shows corresponding profile examples.
Multiple Profile Classes

Texture images have been designed to maximise discrimination at the BMEC boundary. However, capillary images frequently contain regions that have similar texture appearance to endothelial cell material, but lie outside the BMEC boundary. Often, these regions lie close to the boundary, creating locally misleading image evidence. Figure 3.5(b) shows a texture image containing many locally confusing regions lying close to the BMEC boundary. This texture image corresponds to the original image shown in figure 1.3(a). Selecting only profile data that contains intensity step edges will remove profiles that represent the appearance of this confusing image evidence. However, if the profile models do not represent this type of appearance, best fit positions will not lie on the desired boundary. Therefore, a representation of this data should be included in the profile models.

A selection criterion for local image evidence appearance has been designed as follows:

\[
\bar{x}_{\text{innerend}} > \bar{x}_{\text{middle}} + t \& \bar{x}_{\text{outerend}} > \bar{x}_{\text{middle}} + t
\]  

(3.11)

where \(\bar{x}_{\text{innerend}}\) is the average intensity value of the innermost fifth of the profile, \(\bar{x}_{\text{outerend}}\) the average value of outermost fifth and \(\bar{x}_{\text{middle}}\) the average of the central fifth. As above, \(t\) is a threshold value (\(t = 0.75n_s\)). This criterion selects data containing an intensity ‘valley’. Some variation in the width of the valley is allowed by comparing only the inner and outermost data with the central area.

Examples of local image evidence profile appearance chosen with this criterion are shown in figure 3.5(d). They are clearly not drawn from the same linear distribution as the intensity edge profiles shown in figure 3.5(c). We have represented the two classes of profile data as a mixture model of two Gaussian kernels. Each kernel represents each class separately. The prior probabilities of each class of data are estimated by the proportions of data meeting the two selection criteria.
The Mahalanobis distance to a single kernel can no longer be used as a measure of the closeness of a profile model fit to image data (see section 3.2.2). A probability density calculation is now used during model fitting:

\[
P(x) = \sum_i P_i \left\{ \frac{1}{2\pi^{n/2}\sqrt{|S|}} \exp\left( -\frac{M_i(x)}{2} \right) \right\}
\]  

(3.12)

where \( P_i \) is the prior probability of class \( i \), \( n \) is the number of dimensions of the kernel, \( S \) is the covariance matrix and \( M_i(x) \) is the Mahalanobis distance of \( x \) to the class \( i \) (see equation 3.7). The position in the image where \( P(x) \) is maximised, corresponds to the best match of the profile model to image data.

In practice, the covariance matrix \( S \) is often ill-formed, and its determinant evaluates as 0. Profile data are highly correlated and, with small training sets, the true dimensionality is often much less than the number of variables in each vector. This problem can be overcome by using a principal components analysis to reduce the dimensionality of the data. All components with positive, non-zero eigenvalues can be retained, or a proportion of the total variance can be kept. The determinant of \( S \) can now be calculated as:

\[
|S| \approx \prod_i \lambda_i + r
\]  

(3.13)

where \( \lambda_i \) is the \( i \)th non-zero eigenvalue of \( S \). \( r \) is a residual term that applies when non-zero eigenvalues have been discarded. It is usually approximated by the average of the discarded eigenvalues.

Results from an ASM fitting evaluation using Gaussian mixture model profile representation are presented in section 3.3.5, table 3.3.
Image Patch Size

The size of the patch of local image data modelled around each landmark needs to be chosen. The size of the local grey-level model patches will affect the quality of model fitting. The patches must be large enough for the model to be specific to the feature required, yet not be so large that local confusing image evidence close to the boundaries degrades the models.

We have carried out further leave-one-out experiments to determine the optimal patch size. Memory constraints limit the patch area that it is practical to model. Using the training set with 131 annotations, a memory size of 256Mb gives a maximum practical patch area of 360 samples. The patch dimensions we have evaluated range between: $3 \times 120$ to $9 \times 20$ pixels. Results are shown in section 3.3.5, table 3.4.

Total Variance

Another parameter that can be tuned to our application is the total amount of variance that the grey-level patch models represent. Reducing the proportion of the variance that is represented will produces a model with fewer modes which will reconstruct the training data in a less specific manner. However, if the less significant modes of the model are greatly influenced by noise, or misleading data, reducing the variance in the model to exclude these modes may increase the performance of search.

Again, we have carried out leave-one-out search experiments to determine the effect of varying the total variance in the grey-level patch models. The experiments use a normalised, combined model with patch size $5 \times 40$ pixels, trained with selected data. Results for models retaining 66%, 95% and 99.5% of total variance are presented in section 3.3.5, table 3.5.
Figure 3.6: ASM search profile model best fit positions are marked as yellow crosses. The green line marks an expert annotation of the image. (a) shows possible best fit outliers circled in red. (b) shows the PDM reconstruction of the fit positions as a red line.

3.3.3 Model Parameter Estimation

The image information around the BMEC and LEC boundaries is highly variable. In many images there are also significant regions of locally confusing appearance close to the BMEC boundary. These factors can cause profile model best fit positions to be found some distance from the desired boundary. However, given a good initialisation, the majority of the profile model matches lie close to the annotated boundary position. Erroneous profile matches are usually outliers to the general trend.

Figure 3.6 shows the best fit positions for each landmark point at an iteration of ASM search. The majority of the points lie on a roughly consistent boundary, but a few differ for this trend significantly. Possible outliers (profile best fit positions that have a point-to-boundary distance > 20 pixels from the green annotation boundary) have been circled in red.

When calculating PDM parameters for a list of landmarks, each point has equal importance. If some landmarks deviate from the general trend, the whole shape is influenced. Figure 3.6(b) shows a PDM fit to profile match data. In places, outlying
profile matches have pulled the model fit away from the desired boundary. This has happened despite the reasonable consistency of the rest of the profile matches. Clearly, PDM parameter estimation is not robust when some profile matches are unreliable.

We have chosen two methods to modify the standard PDM parameter estimation algorithm with the aim of increasing the tolerance to small amounts of inconsistent landmark data: a weighted pose estimation scheme and Random Sample Consensus (RANSAC) model fitting [52][51].

**Weighted Pose Estimation**

When finding the best fit position of a set of landmark positions it would be desirable to treat outliers with less weight than consistent points. Outliers are caused by poor or confusing image evidence. Profile model fits to image data at these points may be relatively poor, in comparison to the more consistent majority of matches. The Mahalanobis distance, between a sample and the model, or the probability density at the sample point, both give a measure of the quality of a model fit.

We have used probability density (equation 3.12) to form a vector of weights \( w \), one for each profile match. A high density value corresponds to a good match between model and data, and a low density for a poor image match. This measure is normalised over all profile model matches so all values lie in the range [0...1]. This is used as a relative weighting for each model fit position based on model fit quality.

During ASM search, shape constraints are imposed by fitting a PDM to profile model best fit positions. The PDM representation consists of a vector of shape and pose parameters. The first stage in PDM parameter calculation is the alignment of the candidate landmark points to the model mean. This is done as part of the pose estimation process. The relative vector of profile match weights can be used directly in this alignment process. A generalised Procrustes alignment [59] is performed,
minimising the following weighted distance measure instead of the standard squared distance:

\[ d_s = \sum_i w_i ((\bar{x}_i - x_i)^2 + (\bar{y}_i - y_i)^2) \]  

(3.14)

where \( w_i \) is the weight of the \( i \)th profile match, \( \bar{x}_i, \bar{y}_i \) are the coordinates of the \( i \)th model mean landmark, and \( x_i, y_i \) are profile model match position coordinates.

Weighting the alignment in this way provides a method of reducing the influence of poor matches to image data. It is hoped that these poor matches will correspond to outliers, such as those shown in figure 3.6. However, shape parameter estimation remains un-weighted so the effect will be subtle. To evaluate the effect on model fitting, we have carried out a further leave-one-out experiment, comparing an un-weighted scheme with our modifications. Results are given in section 3.3.5, table 3.6.

**Random Sample Consensus (RANSAC)**

The previous section described a method of reducing the effect of outliers on the pose estimation stage of model fitting. However, it is more complex to directly apply a weighing scheme to shape parameter estimation. As an alternative to this, Random Sample Consensus (RANSAC) model fitting [51][52] has been implemented as an alternative method of shape parameter estimation. RANSAC is a method of fitting models to noisy data, that can be tuned to be insensitive to outliers.

The RANSAC algorithm involves randomly selecting a subset of data to use to estimate model parameters. If a large proportion of the original data is consistent with this parameter estimate the model fit is accepted. In this case, a consensus is said to have been achieved. If not, a new random subset of the data is selected and the process repeated. In the case of PDM model fitting, RANSAC can be described as
follows:

1. Randomly select subset \((x_r, y_r)\) containing \(N\) points from the set of data \((x, y)\).

2. Generate model parameters \(b_s\) using \((x_r, y_r)\).

3. From \(b_s\) generate the complete model instance \((x_m, y_m)\).

4. Calculate the distance between each point of \((x, y)\) and the boundary \((x_m, y_m)\).

5. Calculate the proportion of points \(p_d\) with \(d_i < D\).

6. If \(p_d \geq P\) end, returning \(b_s\).

7. Otherwise repeat from step 1.

This algorithm contains the following tunable parameters:

1. \(N\) – the number of points to use in the estimation of model parameters \(b_s\).

2. \(D\) – a distance threshold to determine point to point consistency.

3. \(P\) – the proportion of consistent points that makes up a consensus.

The selection of the subset of points used to estimate model parameters can be done completely at random, or in a weighted roulette scheme. As shall be seen, the results for weighted pose estimation (section 3.3.3, table 3.6) indicate that a low weight for a profile model fit does not indicate that the match is an outlier. In fact, we observe that many confusing profile model matches are highly weighted. For this reason, an entirely random selection method has been implemented, where every point is equally likely to be included in the subset.

*For estimating model parameters using a sub-set of data see chapter 4, section 4.4.4.

†Point-to-boundary distance is described in section 3.3.4.
Figure 3.7 shows an example of the RANSAC fitting process. Four possible PDM fits are shown for the same profile model fit positions. In each case, a different subset of the points have been used to estimate the model parameters $b_s$. The different subsets result in different model parameters and varying proportions of consistent data. The profile model fit positions $(x, y)$ are marked as yellow crosses, the subset of points used to estimate model parameters $(x_r, y_r)$ are marked as green stars. The resulting model fit is shown in red. All points falling within the distance threshold $D$ of the model fit are marked with red circles. Each example has a different proportion of points that are consistent with the model fit, ranging from 0.58 (figure 3.7(a)) to 0.8 (figure 3.7(d)).

One point that has not been addressed here is how a set of model shape parameters can be generated from an incomplete set of point data $(x_r, y_r)$. Briefly, a data imputation process is used to complete the set which can then be processed in the standard manner. The imputation process we use is described in detail in chapter 4. A scheme to calculate shape model parameters from incomplete data is described in section 4.4.4.

The RANSAC process selects a subset of the points that have no influence on the shape parameterisation. If the parameterisation is consistent with most of the data, the fit is accepted. This model fitting process is much more tolerant to data with outliers than the standard PDM method, which should lead to better overall search results. We have carried out a further leave-one-out experiment, to evaluate RANSAC model fitting performance. For this, have chosen the values of $N = 30$, $D = 20$ and $P = 0.9$. These parameters allow 10% of points to be ignorable outliers, and use 60% of the model fit positions to estimate model parameters. If after 20 iterations of RANSAC fitting, a 90% consensus has not been achieved, the parameterisation with the largest consensus set is used. The results of the evaluation are shown in section 3.3.5, table 3.7.
Figure 3.7: RANSAC model fitting. Profile model best fit positions are shown as yellow crosses, the subset of points used to calculate shape model parameters are shown as green pentagrams and the resulting model fit and consensus set as a red line and circles. 30 of the 50 landmark points were used to calculate the model fit in each case on the same set of model fits.
Chapter 3. Modelling Shape and Appearance

3.3.4 Evaluation Measure

Model fit positions have been evaluated against all annotations of the corresponding image. We have calculated the distance of each point on a segmentation boundary to the nearest annotated boundary on the image. As the annotation and segmentation landmark points have no physical meaning along the boundary, only the point–to–boundary distance has been calculated. This measure is illustrated in figure 3.8. The minimum distance between a point $p$ and a line $l$, lies along the normal to the line, $N_l$, that passes through $p$. The distance between the intersection of the line and this normal (point $q$) and $p$ is the shortest point–to–line distance. This measure is represented in figure 3.8(a).

For our purposes, a boundary is represented piecewise by bounded linear sections. The shortest distance between a point $p$ and a bounded linear section $l_b$ is given by the distance between $p$ and $q'$, where $q'$ is the closest point to $q$ that lies on the linear section $l_b$. This distance is represented in figure 3.8(b). When $q$ lies on boundary section $l_b$, $q' = q$ and $d' = d$.

If a boundary is represented by $n$ linear sections, we can extract $n$ point–to–section...
distances for each segmentation landmark. The point-to-boundary distance for a landmark is defined as the minimum point-to-section distance. This measurement is given in pixel units. We have considered the closest annotation boundary to be the ‘best’.

The point-to-boundary distance is calculated for every point on the segmentation boundary to every annotation of the same image. For each point, the final distance is taken to be the minimum distance over the set of annotations.

### 3.3.5 Evaluation Results

This section presents results of evaluations of the factors affecting the performance of ASM search outlined in section 3.3.2. The results of each evaluation have been produced by a series of leave-one-out ASM search experiments.

Each evaluation consisted of 10 image searches. Every search results in a set of point-to-boundary distances. Statistics, from the combined set of point-to-boundary distances for each search experiment, are summarised in tables. Each table gives the overall average point-to-boundary distance $\mu_{\text{all}}$ and the standard deviation $\sigma_{\text{all}}$. Summaries of the best achievable accuracy and the magnitude of gross failures are also presented. The average of the smallest 25% of distances, $\mu_{\text{acc}}$, gives a measure of accuracy of successfully located model points. The average of the largest 25% of distances, $\mu_{\text{rob}}$, gives a measure of the magnitude of the largest failures, and hence the robustness of the model fits. In all cases, lower values indicate model fits that are closer to annotation data.

In tables 3.1 the values of $\mu_{\text{all}}, \sigma_{\text{all}}, \mu_{\text{acc}}$ and $\mu_{\text{rob}}$ are shown for the initialisation position. Initialisation is very close to the annotation for each image, giving a sub-pixel accuracy for each landmark point with $\mu_{\text{all}} = 0.824$ and $\sigma_{\text{all}} = 0.7938$. The initialisation positions remain constant for every experiment as the shape model parameters are not changed.
In every case, local refinement of the starting positions has resulted in model fits that are further away from annotation data. Capillary images are extremely complex and variable. They contain many regions of misleading local image evidence. These factors cause the best local fit position for ASM search to lie away from the desired boundary. The approaches to optimising model building parameters and fitting methods presented in sections 3.3.2 and 3.3.3 can bring the best fit position closer to the annotation data. Overall improvements will be achieved as a series of small increments. The complexity of the image data is such that sub-pixel accuracy and 100% robustness is not achievable for capillary images.

Normalisation

Table 3.1 shows a comparison of ASM search using standard profile models against search using normalised profile models.

<table>
<thead>
<tr>
<th></th>
<th>$\mu_{all}$</th>
<th>$\sigma_{all}$</th>
<th>$\mu_{acc}$</th>
<th>$\mu_{rob}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialisation</td>
<td>0.8240</td>
<td>0.7938</td>
<td>0.3482</td>
<td>1.3743</td>
</tr>
<tr>
<td>Standard Models</td>
<td>17.2629</td>
<td>15.5377</td>
<td>2.2132</td>
<td>39.3760</td>
</tr>
<tr>
<td>Normalised Models</td>
<td>10.6486</td>
<td>11.0986</td>
<td>1.0143</td>
<td>26.4244</td>
</tr>
</tbody>
</table>

**Table 3.1:** Comparison of model fitting performance using standard and normalised grey-level profile models. $\mu_{all}$ and $\sigma_{all}$ are the mean and standard deviation of the point-to-boundary distances between each model fit landmark and the nearest annotated boundary respectively, $\mu_{acc}$ is the average of the smallest 25% of distances, giving a measure of the accuracy of successfully located model points. $\mu_{rob}$ average of the largest 25% of distances, giving a measure of the size of the largest failure, and hence model fitting robustness.

Results from both standard and normalised profile models are substantially worse than their initial positions. Normalising profile data substantially improves model fitting performance on every measure presented in table 3.1. Overall performance is improved from: un-normalised $\mu_{all} = 17.2629$, to normalised $\mu_{all} = 10.6486$. The
accuracy of successfully located landmarks and the robustness of the model fits have also been improved. Profile data normalisation is an important method by which model fitting can be improved.

**Combined Profile Model**

Table 3.2 shows a comparison of ASM search using normalised profile models against search using a normalised profile model built using combined statistics from each landmark position. The combined model search results improve on the normalised model search results slightly. Average fit distance decreases to $\mu_{all} = 10.4134$. The most noticeable improvement has been made to the accuracy of successfully located landmarks. Accuracy ($\mu_{acc}$) has improved by 27% from 1.0143 to 0.7413 pixels to the nearest annotation boundary. These measurements indicate that when good image data is available, using combined profile statistics improves accuracy. Where image evidence is more confusing, the combined model does not improve performance.

<table>
<thead>
<tr>
<th></th>
<th>$\mu_{all}$</th>
<th>$\sigma_{all}$</th>
<th>$\mu_{acc}$</th>
<th>$\mu_{rob}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialisation</td>
<td>0.8240</td>
<td>0.7938</td>
<td>0.3482</td>
<td>1.3743</td>
</tr>
<tr>
<td>Normalised Models</td>
<td>10.6486</td>
<td>11.0986</td>
<td>1.0143</td>
<td>26.4244</td>
</tr>
<tr>
<td>Combined Model</td>
<td>10.4134</td>
<td>11.4445</td>
<td>0.7413</td>
<td>26.9616</td>
</tr>
</tbody>
</table>

**Table 3.2**: Comparison of model fitting performance using normalised and combined grey-level profile models.

**Data Selection and Multiple Class Modelling**

Table 3.3 presents results from ASM searches using profile models that represent different training data. The first row in the table shows results from profile models built with combined statistics of all available profile data. The second row shows results using profile models trained with data that passed the selection criterion described in section 3.3.2. This selection criterion was designed to include only profiles...
with a significant step edge in intensity values. The third row of the table shows results from searches using a mixture model to represent two separate classes of profile data. The first class was the intensity edge profiles, and the second represented the appearance of locally confusing image evidence lying close to the BMEC boundary.

<table>
<thead>
<tr>
<th></th>
<th>$\mu_{\text{all}}$</th>
<th>$\sigma_{\text{all}}$</th>
<th>$\mu_{\text{acc}}$</th>
<th>$\mu_{\text{rob}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Profile Data</td>
<td>10.4134</td>
<td>11.4445</td>
<td>0.7413</td>
<td>26.9616</td>
</tr>
<tr>
<td>Selected Profile Data</td>
<td>7.4661</td>
<td>8.6575</td>
<td>0.5041</td>
<td>20.2911</td>
</tr>
<tr>
<td>Mixture Model</td>
<td>8.9669</td>
<td>10.6762</td>
<td>0.6637</td>
<td>24.5047</td>
</tr>
</tbody>
</table>

Table 3.3: Comparison of model fitting performance using selected data to train grey-level profile models. The first row shows results using unselected data to train profile models. The second shows results using only selected intensity edge data, and the third shows results achieved by explicitly modelling locally confusing evidence together with intensity edge information using a mixture model.

Selecting profile model training data improves model fitting performance on every measure. The best results are achieved using the intensity edge profile model. In this case, overall average distance to the nearest annotation is improved from: unselected $\mu_{\text{all}} = 10.4134$, to selected $\mu_{\text{all}} = 7.4661$.

Incorporating the appearance of locally confusing image data into the profile models degrades model fitting performance slightly. A mixture model consisting of two Gaussian kernels was used to represent these profiles separately from intensity edge data. Locally confusing image evidence represents approximately 10% of the total number of profiles included in the models. Compared to results for the intensity edge profile models, overall average distance increases to $\mu_{\text{all}} = 8.9669$. This figure still represents a substantial improvement over unselected training data.

The positions of annotations frequently lie close to confusing image evidence. Therefore, the appearance of profile data near these positions must be represented in the profile models. The results in table 3.3 indicate that incorporating this information results in less consistent profile matches. We have shown that fitting PDMs to land-
marks is a process that is sensitive to noisy or outlying data. It is this model fitting process that has limited the performance of image searches in this case. Using a model fitting algorithm that is more tolerant to outlying data will improve overall results. Results presented later in this section illustrate this. They show that explicitly representing locally confusing image evidence improves search performance when error–tolerant RANSAC model fitting is used.

Dimensions

Table 3.4 shows an evaluation of the effects of varying profile model local patch size on ASM search. All searches used a combined, normalised profile model trained with selected intensity edge data. Only the dimensions of the local image patches used to train the model were varied. Results show that there is very little difference between each patch size. This is to be expected, as the patch pixel values are extremely correlated. Profile data vectors typically exhibit true dimensionality far less than the number of variables they contain.

<table>
<thead>
<tr>
<th>Width</th>
<th>Length</th>
<th>$\mu_{\text{all}}$</th>
<th>$\sigma_{\text{all}}$</th>
<th>$\mu_{\text{acc}}$</th>
<th>$\mu_{\text{rob}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>120</td>
<td>7.5769</td>
<td>7.4698</td>
<td>0.7132</td>
<td>18.5465</td>
</tr>
<tr>
<td>1</td>
<td>80</td>
<td>7.3479</td>
<td>8.1239</td>
<td>0.6616</td>
<td>19.3781</td>
</tr>
<tr>
<td>1</td>
<td>40</td>
<td>7.6301</td>
<td>8.3379</td>
<td>0.6538</td>
<td>19.9053</td>
</tr>
<tr>
<td>3</td>
<td>120</td>
<td>7.4487</td>
<td>7.9618</td>
<td>0.7191</td>
<td>19.1389</td>
</tr>
<tr>
<td>3</td>
<td>80</td>
<td>7.4661</td>
<td>8.6575</td>
<td>0.5041</td>
<td>20.2911</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>7.5144</td>
<td>8.3184</td>
<td>0.5777</td>
<td>19.7502</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>9.0576</td>
<td>10.4675</td>
<td>0.8002</td>
<td>23.2961</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>9.6318</td>
<td>9.9704</td>
<td>1.0042</td>
<td>23.4524</td>
</tr>
<tr>
<td>5</td>
<td>40</td>
<td>7.0546</td>
<td>7.9033</td>
<td>0.6278</td>
<td>18.9572</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>7.6252</td>
<td>9.1275</td>
<td>0.5754</td>
<td>20.3122</td>
</tr>
</tbody>
</table>

Table 3.4: Effect of varying profile model patch dimensions on ASM model fitting performance. A normalised, combined profile model trained with selected intensity edge data was used.

The best overall results are achieved using a patch size of $5 \times 40$ pixels. Average
distance to annotation data for this patch size is $\mu_{all} = 7.0546$. Robustness has also been improved compared to the patch size we have been using in previous evaluations, from: $3 \times 80 \mu_{rob} = 20.2911$, to $5 \times 40 \mu_{rob} = 18.9572$. In all future evaluations, a patch size of $5 \times 40$ is used.

### Total Variance

Table 3.5 shows an evaluation of the effects of varying profile model total variance on ASM search. Results are similar for each different amount of total variance included in the profile models. The smallest overall average point-to-boundary distance is achieved by keeping 95% of total variance ($\mu_{all} = 7.0546$). This proportion of variance also gives the best robustness results, but slightly reduced accuracy. There is little to choose between the three values of this parameter. Profile models containing 95% of total variance have been used for future experiments as this proportion produces the lowest average distance to annotation data.

<table>
<thead>
<tr>
<th>Total Variance</th>
<th>No. of Modes</th>
<th>$\mu_{all}$</th>
<th>$\sigma_{all}$</th>
<th>$\mu_{acc}$</th>
<th>$\mu_{rob}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>99.5%</td>
<td>20</td>
<td>7.3072</td>
<td>8.5686</td>
<td>0.5430</td>
<td>19.8526</td>
</tr>
<tr>
<td>95%</td>
<td>5</td>
<td>7.0546</td>
<td>7.9033</td>
<td>0.6278</td>
<td>18.9572</td>
</tr>
<tr>
<td>66%</td>
<td>1</td>
<td>8.3292</td>
<td>10.0311</td>
<td>0.5266</td>
<td>23.2134</td>
</tr>
</tbody>
</table>

Table 3.5: Effect of varying the total amount of variance represented in profile models on model fitting performance. Normalised, combined profile models representing selected training data with patch dimensions of $5 \times 40$ were used.

### Weighted Pose Estimation

Table 3.6 shows a comparison of ASM search performance using standard and weighted pose estimation. Both experiments used normalised, combined grey-level profile models contain 95% of total variance in a training set consisting of selected intensity edge
profiles. The weighted pose estimation appears to degrade quality of the final search results. This indicates that outlying profile match positions are often weighted highly during pose estimation. Weights for each profile match were proportional to the quality of the profile model match to image data. Profile match positions that are outliers from the general trend often match well to image data. For this reason, we can conclude that the quality of profile model match to image data does not give a good indication of the reliability of the match position.

<table>
<thead>
<tr>
<th></th>
<th>$\mu_{all}$</th>
<th>$\sigma_{all}$</th>
<th>$\mu_{acc}$</th>
<th>$\mu_{rob}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Fitting</td>
<td>7.0546</td>
<td>7.9033</td>
<td>0.6278</td>
<td>18.9572</td>
</tr>
<tr>
<td>Weighted Fitting</td>
<td>8.1324</td>
<td>10.3917</td>
<td>0.6123</td>
<td>22.2566</td>
</tr>
</tbody>
</table>

Table 3.6: Model fitting performance using weighted pose estimation.

RANSAC Model Fitting

Table 3.7 shows a comparison of model fitting quality between the ASM search using standard PDM parameter estimation and error-tolerant RANSAC estimation. The RANSAC scheme has been designed to reduce the effect of outlying profile match positions when estimating model parameters.

<table>
<thead>
<tr>
<th></th>
<th>$\mu_{all}$</th>
<th>$\sigma_{all}$</th>
<th>$\mu_{acc}$</th>
<th>$\mu_{rob}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Fitting</td>
<td>7.0546</td>
<td>7.9033</td>
<td>0.6278</td>
<td>18.9572</td>
</tr>
<tr>
<td>RANSAC Fitting Single Class</td>
<td>7.8836</td>
<td>10.6113</td>
<td>0.5190</td>
<td>22.7258</td>
</tr>
<tr>
<td>RANSAC Fitting Multi-Class</td>
<td>6.7858</td>
<td>9.1443</td>
<td>0.4724</td>
<td>19.3352</td>
</tr>
</tbody>
</table>

Table 3.7: RANSAC Model fitting performance. The first row shows results using standard model fitting. The second row shows RANSAC model fitting results using profile models containing selected intensity edge data. The third row used RANSAC fitting with profile models representing both intensity edge and locally confusing appearance data.
The best overall results have been achieved when both intensity edge and locally confusing appearance information was represented by profile models (table 3.7, third row). These results show that modelling locally confusing image evidence can produce a valuable source of information to locate the BMEC boundary. Representing this evidence results in less precise but more accurate profile matches. That is, profile match positions are more variable but give better overall accuracy. However, if this appearance is not included in the profile models, consistent but less accurate match positions result. If an error tolerant model fitting scheme is used, the more accurate but variable profile matches, result in better overall results, despite their lack of consistency.

3.3.6 Summary

The best fit position of an ASM to image data does not lie exactly on an annotation. This section has presented a series of stepwise optimisations and modifications to the ASM fitting process, designed to bring the best fit position closer to annotation data. Capillary images are highly complex and variable and as a result sub-pixel accuracy and 100% robustness are not achieved.

Using combined statistics and carefully selecting training data have been shown to be important steps in improving model fit positions. Both result in more robust profile model statistics. If profile models are built with large quantities of data that is not useful in identifying appropriate boundary positions, model fitting will perform poorly.

When building models of profile appearance, all types of appearance must be represented. If this is not the case, profile model match positions will never lie on the desired boundary. Mixture models have been used to represent to separate classes of profile appearance as separate Gaussian kernels. One class represented profiles with an intensity edge, the other represented appearance caused by locally confusing image evidence lying close to the BMEC boundary. Approximately 90% of the total
number of profiles selected were examples of an intensity edge. The remaining 10% were locally confusing appearance. These proportions were used as prior probabilities for each mixture model kernel.

Frequently profile match positions contain outliers to the general trend, due to the complexity of image structure. Integrating both types of profile appearance caused a greater inconsistency in profile fit positions. Outlying profile matches are often associated with strong local image evidence. The standard PDM parameter estimation algorithm is sensitive to noisy data with outliers. An error-tolerant model fitting scheme was required to estimate model parameters from profile match positions. RANSAC model parameter estimation method has been shown to improve capillary image search performance.

In summary, ASM search performance before optimisation resulted in an average distance to annotation data of $\mu_{\text{all}} = 17.2629$. Each modification presented in this section resulted in a slight improvement on this figure. The best results, achieved by RANSAC fitting of normalised, combined profile mixture models trained with selected data, had an average distance of $\mu_{\text{all}} = 6.7858$. This represents a 61% overall improvement in model fitting and is close to the best performance achievable using profile matching. The significant of this improvement has been tested using the Mann–Whitney and Kolmogorov–Smirnov two sample test. In both cases, there is sufficient evidence to reject the hypothesis that both sets of distances come from the same distribution at the 0.01 significance level.

### 3.4 Fourier Descriptors

BMEC and LEC boundaries do not appear to contain any physically consistent features. As PDMs derive much of their specificity from being able to capture consistent structural variation, they may not provide the most appropriate method of analysis for capillary images. Here we consider an alternative whole–boundary parameteri-
Fourier descriptors allow closed boundaries to be decomposed into parametric representations without requiring physically consistent sample points. The resulting descriptions encapsulate the shape of the entire boundary. The parameterisation does not require objects that are highly structured. It has previously been applied to medical images [22][78][141][151], and so lends itself to application in this case.

There are many different ways of defining Fourier descriptors [58][60][136][153], all are based on a Fourier series of periodic basis functions. For example, the function $X(t)$ can be expressed over the interval $(a, b)$ in terms of the basis function $\phi_k(t)$ as follows:

$$X(t) = \sum_{k=1}^{\infty} p_k \phi_k(t)$$  \hspace{1cm} (3.15)

where:

$$p_k = \int_a^b X(t) \phi_k(t) \delta t$$  \hspace{1cm} (3.16)

In equation 3.15, any basis function, $\phi_k(t)$, can be used. Fourier descriptors use a set of sinusoidal (and therefore periodic) basis functions to decompose a signal. If care is taken in formulation, periodic functions are natural candidates for representing closed boundaries.

The coefficient vector $p$ contains the parameters of the representation. In order to use such representations, the sum must be truncated. In Fourier representations, the higher indexed basis functions correspond to higher frequency spatial variation. If the object to be represented has only finite spatial variation, as is the case for most real objects, the expansion can be truncated while still accurately representing the required shape. Giardina and Kuhl [58] defined a bound on the error in representing
a contour that is inversely proportional to the number of terms in the summation.

Zahn and Roskies [153] define descriptors that are invariant to translations, rotations and changes of scale. However, truncating their descriptors can lead to non-closed curves. Granlund [60] defined descriptors that only described closed curves, which he applied to character recognition. Persoon and Fu [112] give a review of some different descriptors, together with their advantages and disadvantages. They used these Fourier techniques to describe and recognise the outlines of handwritten characters and machine parts.

### 3.4.1 Invariant Fourier Descriptors

Staib and Duncan [136][137] proposed a method of whole-boundary description using an elliptic Fourier decomposition. Their parameterisation is invariant to translation, rotation and scale. These features allow the model to extract any significant structure in example boundaries, even when they are not positioned, aligned or scaled consistently. Each boundary only has one unique Fourier representation and only closed boundaries can be represented. These properties represent significant advantages over decompositions mentioned in the previous section.

In the Staib and Duncan parameterisation, closed boundaries are represented by two periodic functions of $t$, where $t$ varies from $[0 \ldots 2\pi]$, $x(t)$ and $y(t)$, which can be represented by their Fourier expansions in matrix form as:

$$
\begin{bmatrix}
x(t) \\
y(t)
\end{bmatrix} = 
\begin{bmatrix}
a_0 \\
c_0
\end{bmatrix} + \sum_{k=1}^{\infty} \begin{bmatrix}
a_k \\
b_k \\
c_k \\
d_k
\end{bmatrix} \begin{bmatrix}
\cos kt \\
\sin kt
\end{bmatrix}
$$

(3.17)

where:

$$
a_0 = \frac{1}{2\pi} \int_{0}^{2\pi} x(t) \delta t \\
c_0 = \frac{1}{2\pi} \int_{0}^{2\pi} y(t) \delta t \\
a_k = \frac{1}{\pi} \int_{0}^{2\pi} x(t) \cos kt \delta t \\
b_k = \frac{1}{\pi} \int_{0}^{2\pi} x(t) \sin kt \delta t \\
c_k = \frac{1}{\pi} \int_{0}^{2\pi} y(t) \cos kt \delta t \\
d_k = \frac{1}{\pi} \int_{0}^{2\pi} y(t) \sin kt \delta t
$$

(3.18)
Chapter 3. Modelling Shape and Appearance

Figure 3.9: A boundary formed by the summation of rotating elliptic phasors, with components shown at three different times, \(t_1, t_2\) and \(t_3\). \(c_1, c_2\) and \(c_3\) are the centres of each ellipse.

The boundary description from this decomposition consists of a list of parameters \(p = (a_0, c_0, a_1, b_1, c_1, d_1, \ldots)\). The first two elements of this vector, \(a_0\) and \(c_0\), contain the coordinates of the centre of gravity of the shape, which provides invariance to translation, as no other parameter is dependent upon the overall position of the shape.

Invariance to rotation, scale and starting point is achieved by considering each term in the summation (equation 3.17) as an ellipse. In this way, the contour can be thought of as a sum of rotating phasors, each individually defining an ellipse and rotating with speed proportional to its harmonic number \(k\). Figure 3.9 shows a boundary being built up as the sum of rotating harmonic ellipses.

A rotating elliptic phasor can be described by the lengths of its major and minor axes, the angle between the x-axis and the major axis, and a phase shift. Phase shift is defined as the phase from the major axis to the position of \(t = 0\). These ellipse parameters can be derived from the raw matrix elements.

Further refinement of this parameterisation is made to produce parameters that are invariant to boundary orientation and scale, as follows:
An ellipse with its major axis aligned with the x-axis with no phase shift can be represented as:

\[
\begin{bmatrix}
  A & 0 \\
  0 & B
\end{bmatrix}
\] (3.19)

where \( A \) and \( B \) are the major and minor semi-axis length respectively. The phasor moves anti-clockwise for positive \( B \) and clockwise for negative \( B \). A rotation of \( \theta \) for this ellipse about its centre of gravity and a phase shift of \( \phi \) can be written as:

\[
\begin{bmatrix}
  \cos \theta & -\sin \theta \\
  \sin \theta & \cos \theta
\end{bmatrix}
\begin{bmatrix}
  A & 0 \\
  0 & B
\end{bmatrix}
\begin{bmatrix}
  \cos \phi & -\sin \phi \\
  \sin \phi & \cos \phi
\end{bmatrix}
\] (3.20)

This expands to:

\[
\begin{bmatrix}
  A \cos \theta \cos \phi - B \sin \theta \sin \phi & -A \cos \theta \sin \phi - B \sin \theta \cos \phi \\
  A \sin \theta \cos \phi - B \cos \theta \sin \phi & -A \sin \theta \sin \phi - B \cos \theta \cos \phi
\end{bmatrix}
\] (3.21)

which represents a general ellipse. The ellipse parameters can be found by solving the following equations for \( A, B, \theta \) and \( \phi \):

\[
\begin{align*}
  a &= A \cos \theta \cos \phi - B \sin \theta \sin \phi \\
  b &= -A \cos \theta \sin \phi - B \sin \theta \cos \phi \\
  c &= A \sin \theta \cos \phi - B \cos \theta \sin \phi \\
  d &= -A \sin \theta \sin \phi - B \cos \theta \cos \phi
\end{align*}
\] (3.22)

Therefore, from equation 3.22:
\[ A^2 = \frac{\alpha + \sqrt{\alpha^2 - 4\beta^2}}{2} \]
\[ B^2 = \frac{2\beta^2}{\alpha + \sqrt{\alpha^2 - 4\beta^2}} \]
\[ \theta = \arctan \frac{Ac + Bb}{Aa - Bd} \]
\[ \phi = \arctan \frac{Ba - Ad}{Ac + Bb} \]

where:

\[ \alpha = a^2 + b^2 + c^2 + d^2 \]
\[ \beta = ad - bc \]

A sign convention, taking \( A \) to be positive and \( B \) to agree in sign with \( b \), is adopted to ensure consistency. The boundary is now represented by the set of refined parameters \( p_{ref} = (a_0, c_0, A_1, B_1, \theta_1, \phi_1, \ldots) \). Scale invariance can be achieved by isolating a single parameter that determines overall scale. To do this, all axis lengths are normalised by \( A_1 \):

\[ A_k' = \frac{A_k}{A_1} \text{ for } k \neq 1 \]
\[ B_k' = \frac{B_k}{A_1} \]

The angles \( \theta \) and \( \phi \) can be converted to relative values to allow the isolation of a global orientation parameter and the removal of a global phase shift, which is arbitrary.

\[ \theta_k' = \theta_k - \theta_{k-1} \]

which leaves \( \theta_1 \) as a global rotation parameter. For phase shift:

\[ \phi_k' = \phi_k - k\phi_1 \]
which gives absolute phase shifts $\phi_k^*$, that can be converted to relative values in the same way as in equation 3.26 to give $\phi'_k$.

The parameters $p_{\text{ref}} = (a_0, c_0, A'_0, B'_0, \theta'_1, \phi'_1, \ldots)$ express the boundary in terms of relative ellipse properties and display the required invariance.

Another interesting feature of this representation, and Fourier representations in general, is that they produce a graceful smoothing of the boundary [87] when the Fourier expansion is truncated. The higher frequency components of the shape are contained in the higher indexed elements of the Fourier summation. Figure 3.10 illustrates the smoothing effect that can be achieved by using a small number of harmonics in a boundary parameterisation. The boundaries in figure 3.10 (a), (b), (c) and (d) were formed by reconstructing the boundary marked as a black dashed line using 1, 2, 10 and 50 harmonics respectively.

In this case, as any parameter set will correspond to a closed boundary, this smoothing by truncation also results in a closed boundary.

Worring et al described segmentation experiments based on real and synthetic images using this method of description to add a-priori shape information. Information was stored by associating a probability density function with each of the Fourier parameters, which described the likelihood of the shape. Each parameter was modelled as an independent Gaussian, with parameters learned from marked-up examples. A local gradient descent algorithm was used to maximise edge response and shape likelihood to segment both synthetic and real images with good results. A wide variety of medical structures, in complex images, were segmented using this technique of adding a-priori shape information. These included the corpus callosum in the brain and the left ventricle of the heart [151].

Staib and Duncan extended their method to 3-D [138] as a general boundary finding system for medical images. The parameterisation proposed is able to represent a limited number of objects, including tori, open and closed surfaces and tubes. Brech-
Figure 3.10: The smoothing effect of truncating the Fourier summation. Fourier reconstruction of the boundary marked as a dashed black line is shown as a heavy blue line. The Fourier parameterisation used 1, 5, 10 and 50 harmonics shown in (a),(b),(c) and (d) respectively.
buhler et al [15] proposed a method of 3-D description based on spherical harmonics. Their technique removed some of the expressive limitations of the Staib and Duncan extension, to allow arbitrarily shaped, simply connected objects to be represented. They demonstrated their technique by modelling various synthetic objects.

Kelemen et al [77] extended the modelling technique to include a principal components analysis of the covariance of Fourier harmonic parameters. In order to improve search they added 1-D grey-level profiles through surface normals. These profiles are evenly spaced based on an icosahedron subdivision of the surface. These two enhancements closely follow the work of Cootes et al [32] on Active Shape Models. Szekely et al [141] used these techniques to model and segment the corpus callosum from MRI volume data. Their model was based around an anatomically defined reference coordinate system that allowed the model to include distributions of spatial position and orientation. The combination of statistical modelling of the FD parameters to impose constraints during search and the intelligent local search technique of profile models are a worthwhile extension to the Fourier modelling technique. We have chosen to implement a 2-D version of this algorithm for application to our capillary images.

Kelemen et al [78] achieved segmentation of cell nuclei from 3-D confocal laser microscopy images using 3-D Fourier models. This work also addressed the problem of the initialisation of Fourier models. Due to local minima in the parameter space, gradient based search techniques will not succeed unless the initialisation of the model is close enough to the desired segmentation. In this case, an accurate initialisation was gained by using a 3-D wave propagation technique to find cell centres. Neueneschwander et al [104] proposed a method of accurate model initialisation from a few user-placed seed points, called Velcro surfaces. Both of these initialisation techniques are problem specific, and not useful in our 2-D analysis task.
3.5 Application of Fourier Descriptors to Capillary Images

Models of shape and appearance are used to impose constraints on image search. To achieve this, the distributions of the Fourier parameters of the training set must be modelled. This can be done directly, using an independent Gaussian model to represent each of the parameters distributions separately. We have chosen to follow the work of Kelemen et al [77] and Szekely et al [141] and apply a principal components analysis to the covariance matrix of the parameters (section 3.5.1). A model fitting algorithm similar to Kelemen’s has been implemented. This technique is based on ASM profile models and shape constraints and is described in section 3.5.2. We have found that the PCA severely restricts the utility of the Fourier descriptor modelling technique in our application and present an analysis of the effects of PCA on model compactness, accuracy and fitting when applied to Fourier parameterisations of BMEC boundary shape.

This section contains descriptions of the PCA representation of Fourier parameter variation, and the model fitting algorithm that has been developed. Evaluations of the modelling scheme and search results are described, and results are presented collectively in section 3.5.3.

3.5.1 Principal Components Analysis of Fourier Parameters

We can carry out a PCA of the Fourier parameterisation in the following manner:

The mean of the parameterisation data of the training set, \( \bar{p} \), with \( s \) examples and \( n \) dimensions, is given by:

\[
\bar{p} = \frac{1}{s} \sum_{i=1}^{s} p_i
\]  

(3.28)
The \( n \times n \) covariance matrix, \( S \), of the data is:

\[
S = \frac{1}{s - 1} \sum_{i=1}^{s} (p_i - \bar{p})(p_i - \bar{p})^T
\]  

(3.29)

The eigenvectors of \( S \), \( e_k \) \((k = 1, \ldots, 2n)\), ordered by non-increasing eigenvalue \( \lambda_k \), are the principal components of the covariance matrix. In a PDM context, the principal components are known as modes, and we used the same terminology here.

Individual parameterisations can now be represented in terms of this set of orthogonal modes in a manner identical to that described in equations 3.5 and 3.4. As PCA parameters are assumed to be normally distributed we can apply shape constraints by limiting their values to lie within a specified number of standard deviations, again in the same way as described in section 3.2.1.

The motivation for applying a PCA in this way is to increase the compactness of the model. This is done by discarding a small proportion of the variance in the training set by truncating the number of modes used in the representation (see section 3.2.1). Reducing the total variance represented by the model will inevitably reduce the accuracy with which the model represents its training set.

Reconstruction error can be used to evaluate the effect of reducing the number model modes. In general, the smaller the reconstruction error for a certain number of modes, the better the model. We wish to have as compact a model as possible, so we require a model with a small number of modes and a low reconstruction error.

To evaluate the effects of truncating the number of modes on reconstruction error, the following experiment has been carried out. 131 annotations were parameterised using a number of harmonics ranging from 10 to 50. A PCA was carried out on the resulting parameterisations varying the proportion of total variance retained from 100\% down to 66\%. The mean point-to-boundary distance over the entire training set was calculated and the number of modes recorded. The point-to-boundary distance measure is described in detail in section 3.3.4. Evaluation results are presented in
3.5.2 Model Fitting

A model fitting algorithm closely following that of ASMs has been implemented. One difference is that the grey-level profile models can not be associated with specific landmark points, as these do not exist in the Fourier parameterisation. The approach of building a combined profile model for each boundary has already been utilised for ASM fitting, so the same technique is used with the Fourier model. Instead of performing a local profile model search at each landmark position, the search is carried out at 50 evenly spaced points around the boundary. The best fit positions of these searches can then be used to generate a new shape parameterisation and the algorithm continues by restricting shape parameters exactly as for ASMs.

The quality of the final model fit is dependent not only on the search algorithm and initialisation, but also on the quality of the model. Figure 3.11 shows model positions during one equivalent iteration of image search for both an ASM and Fourier model. In each of the sub-figures, the yellow crosses represent the best fit positions for the local profile model searches. The green line is the position of a PDM fitted to this data and the red line shows the equivalent position for a Fourier model. The figure shows the effect of reducing the total variance in the model between 100% and 66%. With 100% of the training set variance included in the models, the model positions are very close to profile match positions. As the total variance proportion is reduced, the position of the Fourier model deviates substantially from the profile match positions. The PDM position shows less deviation, and remains close to the bulk of the consistent points. This effect can be seen most clearly in figure 3.11 (d) where only 66% of total variance is included in the model.

This effect has been quantitatively evaluated by running two model fitting experiments. Two models were built, one PDM and one Fourier model, both using 131 BMEC annotations to train the shape and combined profile models. The Fourier
model was built using 50 harmonics representing each boundary. A parameterisation using 50 harmonics results in a vector of 202 elements representing each example. Insufficient training examples are available to calculate robust statistics of a longer boundary representation. Model fits were initialised at the best fit of the model to an annotation of the search image. Each search consisted of five iterations of the models associated fitting algorithm with a search range of 40 pixels in the normal direction and 3 pixels in the tangential direction for the profile models. Results are given in section 3.5.3, table 3.10.

**Figure 3.11:** A comparison ASM and Fourier model fitting. Yellow crosses show profile search best fit positions. The green line shows the PDM instance best fitting the profile search results, and the red line shows the equivalent Fourier model instance. Both models contained 100%, 99.5%, 95% and 66% of total variance in figures (a),(b),(c) and (d) respectively.
3.5.3 Evaluation Results

Here we present the results of the evaluations presented in the previous section. The first evaluation shows the detrimental effect of reducing the total amount of variance in the Fourier model, and is compared to the same evaluation for a PDM. Model fitting results are also given, also indicating the relatively poor performance of the Fourier models.

Principal Components Analysis of Fourier Parameters

Table 3.8 shows the mean training set reconstruction error for Fourier models. The models were built with varying numbers of harmonics and varying proportions of total variance. The smallest reconstruction error for the Fourier model (0.0351) is achieved with 50 harmonics and 100% of variance as expected. However, this model contains 130 modes. The largest reconstruction error (11.6074) occurs with the smallest number of harmonics (10) and the lowest proportion of variance (66%), again as expected.

Table 3.9 shows the mean reconstruction error for various PDM models. The values were obtained using identical training data to those in table 3.8. We can see that the smallest reconstruction error here is 0. This must be true by the definition of the PDM modelling technique. This model is more compact (100 modes) than the equivalent Fourier model and has a better representation of its training set. As the total variance in the model is decreased the reconstruction error increases, however, this increase is substantially less than of that of Fourier models. In the case of Fourier models, anything less than 100% of variance results in a large degradation in reconstruction. Every PDM model is more compact than the any of the equivalent Fourier models and has a lower reconstruction error.
Table 3.8: Effect of varying Fourier model building parameters on reconstruction error and number of modes. Reconstruction error is defined as the mean point to line distance between 50 evenly spaced points on the reconstructed boundary and the original annotation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>66%</td>
<td>6</td>
<td>11.6074</td>
</tr>
<tr>
<td>10</td>
<td>95%</td>
<td>14</td>
<td>9.6274</td>
</tr>
<tr>
<td>10</td>
<td>99.5%</td>
<td>17</td>
<td>9.1608</td>
</tr>
<tr>
<td>10</td>
<td>100%</td>
<td>38</td>
<td>0.5305</td>
</tr>
<tr>
<td>20</td>
<td>66%</td>
<td>12</td>
<td>11.0057</td>
</tr>
<tr>
<td>20</td>
<td>95%</td>
<td>28</td>
<td>8.9915</td>
</tr>
<tr>
<td>20</td>
<td>99.5%</td>
<td>36</td>
<td>8.3618</td>
</tr>
<tr>
<td>20</td>
<td>100%</td>
<td>78</td>
<td>0.0959</td>
</tr>
<tr>
<td>50</td>
<td>66%</td>
<td>17</td>
<td>11.4105</td>
</tr>
<tr>
<td>50</td>
<td>95%</td>
<td>53</td>
<td>7.7160</td>
</tr>
<tr>
<td>50</td>
<td>99.5%</td>
<td>80</td>
<td>5.5550</td>
</tr>
<tr>
<td>50</td>
<td>100%</td>
<td>130</td>
<td>0.0351</td>
</tr>
</tbody>
</table>

Table 3.9: Effect of varying PDM building parameters on reconstruction error and number of modes. Reconstruction error is defined as the mean point to line distance between the landmarks on the reconstructed boundary and the original annotation.

<table>
<thead>
<tr>
<th>Total Variance</th>
<th>No. Modes</th>
<th>Mean Recon. Err.</th>
</tr>
</thead>
<tbody>
<tr>
<td>66%</td>
<td>2</td>
<td>5.7863</td>
</tr>
<tr>
<td>95%</td>
<td>13</td>
<td>2.2667</td>
</tr>
<tr>
<td>99.5%</td>
<td>28</td>
<td>0.5569</td>
</tr>
<tr>
<td>100%</td>
<td>100</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Model Fitting

Table 3.10 shows the results of the evaluation on Fourier model fitting performance. We can see that the results from all of the model fitting experiments are significantly worse than any of the ASM results presented in section 3.3.5. The degradation of performance between 100% and 99.5% of total model variance is much larger than the trend shown in table 3.5.

<table>
<thead>
<tr>
<th>Total Model Variance</th>
<th>$\mu_{all}$</th>
<th>$\sigma_{all}$</th>
<th>$\mu_{acc}$</th>
<th>$\mu_{rob}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>18.2864</td>
<td>19.7064</td>
<td>12.0830</td>
<td>27.6503</td>
</tr>
<tr>
<td>99.5%</td>
<td>27.3956</td>
<td>30.5543</td>
<td>14.2517</td>
<td>57.9279</td>
</tr>
<tr>
<td>95%</td>
<td>29.2521</td>
<td>28.1412</td>
<td>16.0987</td>
<td>53.9905</td>
</tr>
</tbody>
</table>

Table 3.10: Fourier Descriptor Model fitting performance. $\mu_{all}$ and $\sigma_{all}$ are the mean and standard deviation of the point-to-boundary distances between each model fit landmark and the nearest annotated boundary respectively, $\mu_{acc}$ is the average of the smallest 25% of distances, giving a measure of the accuracy of successfully located model points. $\mu_{rob}$ average of the largest 25% of distances, giving a measure of the size of the largest failure, and hence model fitting robustness.

3.5.4 Summary

The results presented in the previous section show that the application of PCA on the Fourier parameterisations of our data does not provide a useful analysis. The resulting models are less compact, less accurate and perform much worse than ASMs in model fitting. Experiments included keeping 100% of total variance from the PCA. This is equivalent to using the raw parameterisations. Even without PCA, the parameterisations resulted in poor compactness and reconstruction errors greater than those from corresponding PDMs.

The Fourier parameterisation was evaluated as it appeared to offer a means of removing the reliance on somewhat arbitrary landmark points. It has proved to be
an unnecessary step, resulting in markedly inferior models. There is no benefit parameterising capillary boundaries in this way, as landmark based PDMs significantly outperform the Fourier techniques for our data.

### 3.6 Active Appearance Models (AAMs)

*Appearance Models* were introduced by Edwards *et al* [47] and Cootes *et al* [30] as a method of extending the PDM modelling methodology to represent both shape and appearance information. Modelling the grey-level appearance of nerve capillary structures has been shown to increase the constraints that the model can impose on image interpretation [46]. As the shape models that we have described above only apply weak constraints, this is a potentially useful feature of Appearance Models in the case of capillary images.

An *Appearance Model* is a Point Distribution Model combined with a *Shape-Free Region Model*. The PDM explains variation in shape, whilst the region model explains intensity variation, but with the important step of shape normalisation. Warping the training images to a consistent shape is important in establishing correspondence between the different training images. Warping can be carried out using any convenient warping algorithm; Edwards [46] used a piecewise affine transformation within a Delaunay triangulation of the convex hull of landmark points to warp the each image to a *mean* shape. We have also implemented Bookstein’s [10] Thin Plate Spline warping as a non-linear alternative to this method.

Warping each image to the same *basis* shape leaves us with a shape-free version of each image. Image values are sampled from the shape-free images giving a vector $g$ for each image. The warping stage ensures that there is correspondence between grey-level values of pixels over the training set, thus each element of the grey-level vector corresponds to the appearance at a specific point in each training image.
By applying PCA to this data we obtain a linear model of the form:

\[ g \approx \bar{g} + P_g b_g \]  

(3.30)

where \( \bar{g} \) is the mean normalised grey-level vector, \( P_g \) is a set of orthogonal *modes of grey-level variation* and \( b_g = (b_{g1}, b_{g2}, \ldots, b_{gt}) \) is a set of grey-level appearance parameters.

The shape and appearance of any example can thus be summarised by the vectors \( b_s \) (PDM shape parameter vector from section 3.2.1, equation 3.5) and \( b_g \). Since some correlation between the shape and grey-level variation may exist, a further PCA is applied as follows. For each example we generate the concatenated vector \( b \),

\[
b = \begin{pmatrix} W_s b_s \\ b_g \end{pmatrix} = \begin{pmatrix} W_s P_s^T (x - \bar{x}) \\ P_g^T (g - \bar{g}) \end{pmatrix} \tag{3.31}
\]

where \( W_s \) is a diagonal matrix of weights, \( w_{s1}, w_{s2}, \ldots, w_{st} \), one for each shape parameter, allowing for the difference in units between the shape and grey models. The elements of \( b_s \) have units of distance, those of \( b_g \) have units of intensity, so they cannot be combined directly. The elements of \( W_s \) are determined as follows. Because \( P_g \) has orthogonal columns, varying \( b_g \) by one unit moves \( g \) by one unit. To make \( b_s \) and \( b_g \) commensurate, we must estimate the effect of varying \( b_s \) on the sample \( g \).

To do this we systematically displace each element of \( b_s \) from its optimum value on each training example, and sample the image given the displaced shape. The RMS change in \( g \) per unit change in shape parameter \( b_{si} \) gives the weight \( w_{si} \) to be applied to that parameter in equation 3.31.

PCA is applied to these vectors, giving a further model:

\[ b \approx Qc \]  

(3.32)
where $Q$ are the eigenvectors of the covariance of $b$ over the training set and $c$ is a vector of appearance parameters, $c_1, c_2, \ldots, c_t$, controlling both the shape and grey-level appearance of the model. Since the shape and grey-model parameters have zero mean, $c$ does too.

Also, since the columns of $Q$ are orthogonal, we can obtain $c$ for a given $b$ simply:

$$c = Q^T b$$

(3.33)

### 3.6.1 Model Fitting

Objects represented in this way require a high number of appearance parameters (faces are typically represented by between 50-100 parameters). This high dimensionality of the search space makes conventional search techniques impractical.

To overcome this drawback, Cootes et al [30] and Edwards [46][49] developed a method of optimising model fitting. Following the work of Covell [37] and Black and Yacoob [9], image difference patterns corresponding to changes in each model parameter are learnt and used to modify the model estimate. This association is learned off-line during the training of the AAM in the following way. An assumption is made that the relationship between the model fit error and the error in the model parameters is linear:

$$dc = AdI$$

(3.34)

where $dc$ is the error in the model parameters and $dI$ is the model fit error matrix. $A$ is calculated using multiple multivariate linear regression on a sample of known model displacements. In this way, a-priori information to guide optimisation is calculated as the model is developed. Optimisation is carried out in an iterative manner until convergence is achieved.
3.7 Application of AAMs to Capillary Images

The variability of shape and relative position of the structures in our capillary images make them a challenging data set to represent with AAMs. A landmarking scheme has been devised for use with PDMs (section 3.3.1) which we will again apply. The landmarks must be used, not only to define the shape of the capillaries, but also their appearance region–boundaries and the warps that must be applied to generate a shape free appearance region. We have chosen to model capillary appearance within a convex hull of the BMEC boundary landmarks.

The next section describes an investigation into the choice of an appropriate warping algorithm. We have found the standard affine warping procedure, based on a Delaunay triangulation, is not well suited to this data. A manual triangulation can improve the model, but again is not satisfactory. A thin plate spline warper has been implemented, and provides the best shape free data. However, the performance of AAM model fitting is still found to be very poor. This is caused by distortions that result from the warping of appearance data from extreme shapes. These distortions lead to a poor representation of capillary appearance.

3.7.1 Optimising Image Warping

This section gives details of three methods of warping appearance data to a shape-free mean shape.

Delaunay Triangulation Warper

The standard method of warping used in creating a shape-free region model is an affine transformation based on a Delaunay triangulation [46]. Each training example is warped onto a standard shape, which is usually chosen to be the mean shape of the training set. Figure 3.12 shows a Delaunay triangulation of the mean shape of a
training set containing 23 example images. The training set was chosen manually to include only texture images with a good visible separation of the various structures in the image.

![Figure 3.12: A Delaunay triangulation of the mean capillary shape.](image)

The resulting triangulation is somewhat irregular, with widely varying triangle areas. The mean shape of the training set contains a very constricted segment of the lumen which is not a good feature of a shape-free representation. Due to the variation in capillary shape the triangles, when mapped onto a specific example, often become flipped, i.e. two of the vertices reverse position. The implementation of the warping algorithm leads to the values in the flipped triangles to be marked as invalid. These values propagate throughout the modelling algorithm and cause major distortion in the resulting model. Figure 3.13 shows the first four modes of the appearance model built with the 23 hand picked training examples. Each mode is severely affected by the invalid data introduced by flipped triangles. Although this effect can be minimised by implementation changes, the resulting model indicates that this triangulation is not suitable to be used to warp the capillary images.
Figure 3.13: The first four modes of an appearance model built using a piecewise linear warper, based on a Delaunay triangulation.
Manual Triangulation Warper

A manual triangulation and shape-free scheme has been designed to address the problems encountered with the Delaunay warper. The triangulation is shown in figure 3.14. The shape-free form has been regularised by using the best-fitting ellipses to the LEC and BMEC boundaries rather than the mean shape. This removes unwanted shape deviations from the shape-free form. The triangulation has also been made more coarse, which decreases the likelihood of flipped triangles.

Figure 3.14: A manual shape-free region with triangulation.

The modes of the appearance model built with an affine warper using this manually designed triangulation show significant improvement over the Delaunay model. The amount of invalid data has been reduced and now occurs only towards the extremes of each mode. However, the problem is still present and it is clear that it cannot be completely eliminated using a piecewise warping scheme.
Figure 3.15: The first four modes of an appearance model built using a piecewise linear warper based on a manually defined triangulation.
Chapter 3. Modelling Shape and Appearance

Thin Plate Spline Warper

Bookstein’s [10] thin plate splines provide a method of warping an image with a continuous transform. Briefly, a thin plate spline defines an interpolation between two sets of points, which can be applied at all intermediate points. The function can be thought of as a thin metal plate that has been deformed to map one set of points to the other. The function minimises the bending energy of the plate, providing the smallest deformation that will provide the appropriate mapping. The thin plate spline technique has many other potentially useful properties, a summary of which is given in [12].

Figure 3.16 shows an example thin plate spline warp, mapping one set of capillary landmarks to another. The warp can be applied to intermediate points, shown in figures 3.16(c) and 3.16(d). Figure 3.17 shows the first four modes of an appearance model built using the thin plate spline warper. In this model, although there are no regions of invalid data, there seems to be significant distortion in the appearance of the capillaries, especially towards the extremes of the modes. The large shape variation in the training set causes much of the appearance data to be stretched or swirled across the shape-free region. This effect can be seen in figure 3.16 where the is grid is extremely irregular in many areas. Although this model appears better than the previous two attempts, the distortion creates a misleading model of capillary appearance that will clearly adversely affect search.

3.7.2 Model Fitting

To fit an appearance model to image data, Active Appearance Models learn the relationships between parameter displacements and error patterns [46]. Using this scheme, a poor model of shape-free appearance will result in poor model fitting results. To evaluate the effect of the perceived distortion in our nerve capillary models, we carried out the following model fitting experiments. A series of leave-one-out cross-
Figure 3.16: A thin plate spline warp. (a) shows the model mean shape and landmark points, marked as blue crosses. (b) shows the target landmark points that define the thin plate spline. The regular red grid of intermediate points shown in (c) is warped to the grid in (d). Corresponding points in (c) and (d) are used as sample positions for shape–free appearance.
Figure 3.17: The first four modes of an appearance model built using a thin plate spline warper.
validation experiments were carried out. In each, a search conducted on each image in the training set using a model built using all other images. Models were built using each of the warping techniques outlined above. The regression was performed on data from 50 displacements for each model parameter in every image. The sizes of the displacements were as follows: up to $\pm 10$ pixels in $x$ and $y$, $\pm 0.1$ in $s \cos(\theta)$ and $s \sin(\theta)$ (where $s$ is scale and $\theta$ is orientation) and $\pm 1$ standard deviation of each model parameter. Models contained 99% of shape and appearance variance, with 99% of combined variance. The starting point for each search was the best fit of the model to the original annotation. This is the best possible initialisation position as in section 3.3.5. The searches were run for 20 iterations. The results are reported in table 3.11, using the same point-to-boundary distance measures as previously described in section 3.3.5.

<table>
<thead>
<tr>
<th></th>
<th>$\mu_{all}$</th>
<th>$\sigma_{all}$</th>
<th>$\mu_{acc}$</th>
<th>$\mu_{rob}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialisation</td>
<td>5.8789</td>
<td>4.6676</td>
<td>4.6676</td>
<td>10.4833</td>
</tr>
<tr>
<td>Delaunay Warper</td>
<td>19.7901</td>
<td>23.6707</td>
<td>2.5434</td>
<td>48.7307</td>
</tr>
<tr>
<td>Manual Warper</td>
<td>29.2963</td>
<td>41.4034</td>
<td>2.8746</td>
<td>79.9652</td>
</tr>
<tr>
<td>Thin Plate Spline Warper</td>
<td>36.5553</td>
<td>40.9381</td>
<td>5.4233</td>
<td>88.5161</td>
</tr>
</tbody>
</table>

Table 3.11: Comparison of AAM fitting performance using three different warper types. $\mu_{all}$ and $\sigma_{all}$ are the mean and standard deviation of the point-to-boundary distances between each model fit landmark and the nearest annotated boundary respectively, $\mu_{acc}$ is the average of the smallest 25% of distances, giving a measure of the accuracy of successfully located model points. $\mu_{rob}$ average of the largest 25% of distances, giving a measure of the size of the largest failure, and hence model fitting robustness.

Table 3.11 shows that the inclusion of the extra appearance data degrades the initialisation ($\mu_{all} = 5.89$) compared to that of similar ASM experiments ($\mu_{all} = 0.82$). ASMs represent only shape information, whereas AAMs consist of a combined model of shape and appearance. If significant variance is associated with appearance data, shape information will be discarded when the number of modes of the model are truncated.
The fitting results show significantly worse final point-to-boundary distances for all AAMs than have been achieved under similar conditions using ASM fitting. The search failures are caused by warper inadequacies. The thin-plate spline model appears to have performed substantially worse than the other models. The thin-plate spline model contains no areas of invalid data, as so learns associations between error patterns and parameter changes with greater efficiency than the other two models. The model therefore moves further during each iteration of search. None of the models positions were stable after 20 iterations. If search had been continued results would become progressively worse for each model.

### 3.7.3 Summary

AAMs seem a potentially useful modelling methodology for application to our images. However, the warping process used to produce a shape-free form of each training example is not suited to structures with a large variation in shape. Previous, successful applications of AAMs have involved images with rather limited shape variation, such as faces. This proves to be the factor that limits both the model quality and the model fitting performance of the technique. Due to these problems, AAMs are not a suitable method to use for capillary images.

### 3.8 Summary

Point distribution models have proved useful in the analysis of medical images in many cases [32][31][50][79][132][133]. They provide a good choice of modelling technology when strongly constrained structures are to be represented (e.g. anatomical structures). It was not clear that they would be such a good choice for the highly variable and unstructured nerve capillaries. Two alternative modelling strategies were examined.
Fourier descriptors were an initially promising approach to boundary parameterisation. The Staib and Duncan harmonic ellipse formulation [136] has many desirable characteristics, such as orientation invariance, representing only closed boundaries and graceful shape smoothing. The method is a whole-boundary parameterisation, and so does not require the definition of an ad hoc PDM landmarking scheme. However, investigations into Fourier descriptors of boundary data revealed that resulting models were far less compact than corresponding PDMs. Model fitting performance was found to be substantially less accurate than that achieved using ASMs.

Appearance models include a representation of the grey-level appearance of the entire capillary object. This extra information should have the potential to increase the constraints on image interpretation, and lead to more accurate and robust model fitting. However, it has been shown that warping the highly variable capillaries to a constant reference shape introduces significant errors and distortion. The models that result have been observed to be a poor representation of capillary appearance. Furthermore, AAM search experiments show that these perceived warping artifacts substantially degrade model fitting performance. Before modelling, texture analysis has been applied to the capillary images. The texture measures, described in chapter 5, have been designed to maximise the discrimination between regions on either side of the BMEC boundary. Capillary appearance has already been addressed by this texture modelling step, so explicitly modelling this data may be an unnecessary step.

In conclusion, PDMs have been shown to be a good method of representing capillary shape, even though they exhibit no consistent structure or features. A simple geometric landmarking scheme has been developed to parameterise both the BMEC and LEC boundaries. ASMs have been optimised to provide a method of fitting the PDMs to texture processed image data. The following modifications to the classical ASM have been implemented:

- **Error tolerant RANSAC model parameter estimation.** This scheme reduces the sensitivity of model fitting to inconsistent profile model match positions. Incon-
sistent profile match positions are often caused by complex and variable image evidence. RANSAC parameter estimation has been shown to improve overall model fitting results.

- **Profile model optimisation.** Profile models have been built using combined normalised data from every point on the boundary. The quality of the profile models has been improved by carefully selecting the training set. Locally confusing image evidence lying close to the BMEC boundary has been explicitly included in representation using mixture modelling.

The result of applying these modifications is a significant improvement in potential capillary image segmentation accuracy.
Chapter 4

Modelling Intermittent Features: Structured Point Distribution and Structured Appearance Models

4.1 Introduction

Diabetic nerve capillary structures are highly variable in shape and appearance. Occasionally, the lumen (the tube that runs through the centre of the capillary) becomes so constricted as to be practically unobservable. Figure 4.1 shows two example capillary images. In figure 4.1(a), the lumen is clearly visible. In figure 4.1(b) the structure is very difficult to identify, and a clinician has not been able to annotate it. The position and shape of the lumen when it is present, may contain a valuable source of constraints that can be imposed during image interpretation. Therefore, it is desirable to include this data in a statistical shape model. If lumen data is to be modelled, some method must be chosen to represent intermittently present features.

This chapter describes work to develop models capable of representing features that are only present intermittently in images of a structure. We have done this by extend-
Chapter 4. Modelling Intermittent Features: Structured Point Distribution and Structured Appearance Models

Figure 4.1: Two examples from the set of nerve capillary images. Both BMEC and LEC boundaries are marked as yellow lines where available. (a) shows an example where the lumen is visible, (b) is an example where it is not.

Introducing the PDM [36] paradigm. This is a continuation of the optimisation of PDMs for use with nerve capillaries that has been described in chapter 3. Intermittent features have been included in these models by treating unobservable structures as missing landmark data and imputing expected positions. We have developed a novel method of data imputation that retains weak relationships in the data and minimises bias. Our method evaluates well against other imputation techniques in the literature.

The models described in this chapter are not confined to application on nerve capillary images. Many other structures contain features that are not present in every instance. To illustrate this, we have demonstrated the use of intermittent feature models on artificial data, faces, where we include the moustache as an intermittent feature, and nerve capillaries. Intermittently present features are also common when analysing a stack of histological slice images. Features appear and disappear from slice to slice with possibly arbitrary patterns of emergence.
4.2 Modelling Intermittently Present Data

Several modelling strategies suggest themselves to allow intermittent features to be represented. Separate models could be built: one including only examples where the lumen is present, and one with only examples where it is not. We could build a two-tier hierarchical model, consisting of a model of the basement membrane/endothelial cell material boundary (BMEC) together with an internal model representing the lumen/endothelial cell material boundary (LEC) and some indication of whether the lumen is present. The technique we have chosen is to build a single model and impute the landmark positions of features that are unavailable. This is combined with a labelling of whether the lumen is present or absent in each example.

Each of these proposals has advantages and disadvantages. Taking them in order: it is a simple task to build two separate models to represent the capillary structure. This requires no modification to the PDM algorithm that we have applied to the BMEC structure. However, once this has been done we are left with a very inadequate representation of the nerve capillary. In particular, we have no way of relating the modes of each model to one another in a meaningful way. Corresponding modes could represent quite separate relationships in the data. Each model only represents a subset of the variability found in the shape of the BMEC. The split in how the data relationships are modelled is dependent on how the training set has been split into groups. Neither model represents all variation present in the training data. There is no obvious way to decide which of the models to apply to an unseen image for capillary detection. If both models are applied in every case, there is no straightforward way to distinguish the better of the two resulting fits. The drawbacks of this, the most simple proposal, clearly outweigh any benefits of application.

A two-tier hierarchical model looks more promising. Here, the entire BMEC training set can be represented by the BMEC model and the entire LEC training set by the internal LEC model. However, these models are built independently of one another so do not represent relationships between the shape or position of the two structures.
We cannot overcome this by building a combined model in the same way as we do when building AAMs, as LEC boundary parameterisations are not available in all cases. Although the hierarchical nature of the proposed model adequately represents the structure of the nerve capillary, it may break down when applied to more complex missing data patterns. For example, when looking at histological slices, structures may emerge over a few slices, rather than appearing complete after being entirely absent in the previous slice. We would need to build a hierarchy that could cope with all possible patterns of emergence for each structure to model this data adequately, which may be a complex task.

Finally, if we were able to build a single model out of all the data we would, of course, capture all the relationships present in the data. The model would also represent the entire training set. Furthermore, the structure of missing data can be modelled on a landmark by landmark basis allowing for arbitrary patterns of emergence. These features give us significant advantages over the previous two proposals and it is this method that has been chosen for development. There are, however, some caveats to this proposal. Treating absent features as missing data and imputing their values affects any subsequent statistical analysis. The statistical analysis used by PDMs is a linear PCA of the covariance matrix. Imputed values can bias the estimated covariance matrix by inadvertently introducing false relationships. This effect can be minimised by careful choice of imputation algorithm as will be demonstrated in section 4.3.2.

The usual interpretation of missing data in statistical analysis is that unobserved values have not been measured for some reason. This is not the case for intermittently present features, their values genuinely do not exist in some cases. Missing values must imputed in such a way as to retain relationships in the underlying data distribution.

The modelling strategy that we have developed is represented by the schematic presented in figure 4.2. There are three phases in the scheme. The first is the completion
of the data set by imputation. A standard PDM is built from the completed data set, resulting in shape parameters for each training example. The second phase is to label the structure of the missing data. That is, to record whether each landmark point in the intermittent data was present or absent. This has been done using simple binary vectors indicating the presence/absence of each landmark in each training example. The dimensionality of these vectors is reduced using PCA. The resulting structure parameters are combined with the PDM shape parameters and a further PCA is carried out. This results in a set of parameters representing all the relationships between the shape and structure of the data.

The following section describes a comparison of data imputation techniques that could be used in the first phase of modelling. Details of how the missing data structure is modelled are given in section 4.4.

### 4.3 Data Imputation

The basic idea of data imputation is to estimate missing values using available data. Missing data is a commonly encountered statistical problem and data imputation is
4.3.1 Imputation Methods

Statistical imputation techniques commonly estimate missing values assuming a model of the data distribution [86]. Model parameters are estimated and the model is then used to estimate missing values. This process is illustrated in figure 4.3. The figure shows a 2D Gaussian model representing a data distribution. The model is represented by the mean position (a blue cross) and the covariance ellipse. An incomplete data vector, where only the value of variable 1 is known, is represented as a red cross. Statistical imputation techniques can be used to estimate the missing value of variable 2, which must lie on the dashed line.

The simplest method available is to replace each missing data value with the mean of all the other values of that variable (see equation 4.10). This process is represented
in figure 4.4 for a 2D data example. The value of the estimate for variable 2 of the incomplete data vector is shown as a green cross.

This method is very crude, and will clearly introduce a bias towards the mean of the data. This causes an underestimate of the amount of variance within a training set, so this method is undesirable for use in PDM shape models. We have demonstrated this characteristic in an evaluation described in section 4.3.2.

Dear [42] proposed an imputation technique based on Principal Components Analysis. Firstly means are imputed in the data, then the first principal component is computed. Imputation estimates of missing data are obtained by projecting the mean completed data vector onto the first principle component. Missing data values can then be read off. This process is represented in figure 4.5.

Using this technique, gross trends in the data are better preserved. The principal components associated with the imputed mean values are likely to be discarded unless this is the dominant trend in the data. Again, a bias towards mean values has been introduced, although the effect has been reduced compared to simple mean
imputation. This algorithm has also been subjected to a comparative evaluation outlined in section 4.3.2.

In practice, model parameters are not known in advance. They must be estimated using the incomplete data. Beale and Little [7] present an iterative method to produce a maximum likelihood estimate of the missing values based on the assumption that the data comes from a multivariate normal distribution. The maximum likelihood estimate position is illustrated in figure 4.6.

Completed data sets can be produced by iteratively updating model parameters and re-estimating missing values. This is achieved by iterating through equations 4.1–4.4 until convergence.

\[
\hat{x}_{ij} = E(x_{ij} | P_i; \mu, \Sigma), \quad (4.1)
\]
Figure 4.6: Data imputation using maximum likelihood estimate for 2D data. The maximum likelihood estimate of the value of variable 2 is shown as a large green cross.

\[ \mu_j = \frac{1}{N} \sum_{i=1}^{N} \hat{x}_{ij}, \]  
\[ \sigma_{jk} = \frac{1}{N} \sum_{i=1}^{N} \{(\hat{x}_{ij} - \mu_j)(\hat{x}_{ik} - \mu_k) + \sigma_{jk,P_i}\}, \]  
\[ \sigma_{jk,P_i} = \text{cov}(x_{ij}, x_{ik}|P_i; \mu, \Sigma) \]

where \( P_i \) is the set of variables observed in example \( i \), \( \mu \) and \( \Sigma \) are the mean and covariance of the multi-variate normal model and \( x_{ij} \) is the \( j \)th observed value in example \( i \). Note that

\[ \hat{x}_{ij} = x_{ij}, \text{ if } x_{ij} \text{ is observed}, \]
\[ \hat{x}_{ij} = \text{a linear combination of the variables in } P_i, \text{ if } x_{ij} \text{ is missing.} \]

To begin, initial values of \( \mu \) and \( \Sigma \) are obtained, usually by imputing mean values. At
each iteration, the data are completed by equation 4.1 and new means and covariance matrix found for the variables. The covariance is adjusted by adding $\sigma_{jk,P}$ for every observation $i$ to the $jk$th element. Note that this adjustment is zero unless both $x_{ij}$ and $x_{ik}$ are missing.

These equations can be thought of as a form of Expectation Maximization (EM) algorithm [44], with iterative re-estimation of the underlying distribution leading to new maximum likelihood estimates of the missing data. Before this algorithm can begin, an initial estimate of the missing data must be generated. We have used the mean value of each missing variable over all available data. The algorithm can be extremely sensitive to the quality of this initial estimation as is shown in the subsequent evaluation. Also, in order to compute equation 4.1 it is necessary to calculate the inverse covariance matrix $\Sigma^{-1}$ which is extremely memory or cycle intensive with data of high dimensionality.

To address these issues, we have developed a further method of imputation. The algorithm has been designed to retain weak data relationships in the imputed data. Specifically, we wish to impute values in such a way as to retain relationships found in the original data and do this without reducing the total variance in the data. The algorithm is based on an iterative version of Dear’s PCA imputation with several modifications. The algorithm can be described as an iteration of the following equations until convergence:

\[
(P_{xm}, \mu_x, \sigma^2_{xm}, b_{xm}) = \text{pca}(x, m) \tag{4.5}
\]

\[
\hat{x} = \mu_x + b_{xm}P_{xm}^T \tag{4.6}
\]

\[
x_{ij,M_i} = \hat{x}_{ij,M_i} \tag{4.7}
\]
where \( \mathbf{x} \) is the original dataset, \( x_{ij} \) is the \( j \)th observed value in example \( i \), \( M_i \) is the set of variables missing in example \( i \), and \( \text{pca} \) is a function that computes the first \( m \) principal components \( (P_{xm}) \), the variance each mode represents \( (\sigma_{xm}^2) \) and the mean \( (\mu_x) \) of \( \mathbf{x} \), together with the associated reconstruction parameters \( (b_{xm}) \) for each example.

The algorithm is initialised by imputing means into \( \mathbf{x} \), with contains missing values. Equations 4.5–4.7 are cycled through until convergence. Setting \( m = 1 \) and performing one iteration of this process is equivalent to Dear’s PCA imputation.

We define convergence to have occurred when the maximum absolute difference in the values of \( x_{ij,M_i} \) and \( \hat{x}_{ij,M_i} \) is less then \( 10^{-6} \). Choosing the number of modes \( (m) \) to use in estimating imputed values is crucial to the well-mannered convergence of the algorithm. We use the following scheme:

1. Set \( m = 1 \).

2. Repeat until convergence:
   
   a. Calculate first \( m \) modes (equation 4.5)
   b. Estimate and replace missing data values \( x_{ij,M_i} \) (equations 4.6 and 4.7)

3. Unless \( m \) equals maximum number of modes, set \( m \rightarrow m + 1 \) and return to step 2

Firstly, \( m \) is set to 1 and the algorithm is run to convergence. The imputed data is now consistent with data patterns represented by the first mode of variance, but no others. To include relationships represented by other modes we increase \( m \) by 1 and repeat the convergence, starting at the result of the previous iteration. At each stage of the iteration we are including effects of higher modes in the imputed data, and matching it more closely to the original data patterns. However, the imputed data itself also has some influence on the modes produced by PCA. If we were to continue
to include all the modes we would eventually reach a mode that was dominated by
the effect of the imputed data. In this case, the algorithm would not converge as
the imputed data would be updated to reinforce the mode, increasing its significance.
This mutual reinforcement loop would continue indefinitely and the mode would
eventually dominate all other variance in the data. We must therefore choose a point
at which to stop the iteration. If $p$ is the proportion of examples in the data that are
complete, then we can stop the algorithm using the following criterion:

$$\frac{\sum \sigma^2_{x_m}}{\sum \sigma^2_x} > p$$  \hspace{1cm} (4.8)

where $\sigma^2_x$ is the variance of all modes of $x$. This stopping criterion has been chosen
to ensure that we stop at a mode before any pattern in the imputed data dominates.
The criterion is somewhat heuristic, and has not been shown be be optimal. However
the algorithm, as it is stated here, is shown in section 4.3.2 to perform well compared
to the other three techniques described in this section.

### 4.3.2 Evaluation

A comparative evaluation of some of the prospective methods of data imputation was
carried out to determine their expected performance characteristics within the PDM
paradigm. Beale and Little’s ML algorithm [7], the iterated PCA scheme, Dear’s
PCA based estimation [42] and simple mean imputation schemes were evaluated
using synthetic and real data. Results demonstrate that the ML algorithm performs
well when strong patterns are contained in the data. The method’s performance is
poor when this is not the case. Iterated PCA imputation achieves almost equivalent
results to the ML algorithm when strong relationships are present in the data. The
technique outperforms all the other algorithms when applied to nerve capillary data.
Both single pass PCA estimation and mean imputation perform comparatively poorly
for our purposes in all cases, as expected.
Chapter 4. Modelling Intermittent Features: Structured Point Distribution and Structured Appearance Models

Figure 4.7: Examples from the set of multi-variate synthetic data. Variable value is plotted against variable number, showing an imperfect linear relationship in each example.

Synthetic Data

A matrix of synthetic data was created with the following algorithm:

1. Repeat for each integer value of $i$ between $[1 \ldots 50]$

   (a) Set $x_i = (i, 2i, 3i, \ldots, 10i)$

   (b) Set $x_i = x_i + ir$ where $r$ is a vector of random displacements

This results in a matrix with dimensions $50 \times 10$, containing 50 examples of a 10 element vector. There is one overriding incremental relationship within each vector, proportional to the first value in the vector. This relationship is not perfect, as it has been distorted by a vector of random factors $r$ (between -0.5 and 0.5) scaled by the first element in each vector $i$. Figure 4.7 shows three examples from this dataset.

A proportion of this data is removed at random to produce a set of incomplete data. The data that has been removed can be restored with each of the imputation methods and the resulting completed data analysed.
After imputation, the completed data can be compared to the original data in a number of different ways. We have chosen to display the average Euclidean distance between each example and its imputed value. This gives a measure of raw error introduced by the imputation. In general, the smaller this distance, the more successful the imputation algorithm at completing data with the specific characteristics we have defined.

As we know the underlying relationship in the synthetic data, we can also calculate the distance between the imputed data and the ideal value that we would expect if there had been no random distortion in the training data. This will provide us with a measure of how well the relationship is being preserved through the imputation step. A smaller distance between the ideal data and the imputed data indicates the underlying pattern in the data is being modelled and the random variation discarded.

We test the robustness of our four algorithms by removing increasingly large portions of the data at random. The proportion of data removed was varied from 1% to 50% in steps of 1%. The results for each of our measures are shown in figures 4.8 and 4.9.

The performance of both the EM algorithm and the iterated PCA algorithm is almost equivalent in both measures. Both methods are capable of completing the dataset to a reasonable accuracy, even when large proportions of the data are missing, whilst preserving the underlying pattern in the data. The single pass PCA and mean replacement algorithms both give rise to large differences between the imputed and original data. Therefore, these methods are clearly inadequate in this case. However, the synthetic data we have used in this analysis is based on a very strong relationship which is not the case for the nerve capillary landmark data.

Nerve Capillary Landmark Data

To evaluate the applicability to our specific task, we compare each method of imputation using the nerve capillary landmark data. A subset of data from the capillary
Chapter 4. Modelling Intermittent Features: Structured Point Distribution and Structured Appearance Models

Figure 4.8: The Euclidian distance between synthetic data and imputed data. The proportion of data removed before imputation was varied between 1% and 50%. Iterated PCA and ML curves almost coincide.

Figure 4.9: The Euclidian distance between ideal values of the synthetic data and imputed data. The distance between the ideal data values and the randomly perturbed synthetic data is shown as a horizontal dashed line. Iterated PCA and ML curves almost coincide.
training set has been used in this evaluation. The data consists of the 50 BMEC boundary landmarks from 30 randomly selected nerve capillary examples. Four examples from the set are shown in figure 4.10.

![Figure 4.10: Four examples of BMEC boundary landmark data used to evaluate imputation methods. Landmark points are marked as crosses on each boundary.](image)

To analyse the performance of the algorithms applied to this data, we have used the raw distance measure described above. Since we do not know the underlying relationships in the data, the second method cannot be used in this case. We have developed another evaluation measure for these results. An important feature of any model is that it captures all the significant variation present in its training set. Therefore, it is important that any data imputation does not decrease the total variance in the data significantly. We have calculated the difference in total variance between the imputed data and the original data. A value close to zero indicates that the original variance in the data is being preserved.

The experiments were carried out as before, varying the amount of missing data between 1% and 50%. Results are shown in figures 4.11 and 4.12.

The ML imputation algorithm, now exhibits markedly worse performance characteristics to those produced with synthetic data. The ML algorithm performs much worse than the iterated PCA algorithm and the single pass PCA algorithm in both evaluation measures. Figure 4.11 shows that the ML algorithm now approximates the performance of mean replacement imputation and does not preserve the relationships of the original data. Mean values are used as the initial estimation of missing values.
Figure 4.11: The euclidian distance between nerve capillary landmark data and imputed data.

Figure 4.12: Difference between the total variance in the nerve capillary landmark data and the imputed data.
and the ML imputation method has become swamped by the relationships introduced by taking this step. The genuine relationships in the data are not strong enough to overcome this, and ML imputation now closely approximates the initial estimation of mean values. This leads to a large raw error distance and most importantly, a large underestimate of the total variance in the data which approximates the performance of simple mean replacement.

The iterated PCA algorithm results in the best performance in both measures. Because the algorithm has been tailored to the specific statistical analysis that is to be carried out after imputation, the data retains the characteristics required by this analysis. PCA provides a method of selectively preserving correlations in the data. Missing data estimates are refined to come into line with the relationships in the rest of the data in a controllable manner. This enables even weak relationships to be preserved and the total variance of the data to be restored closer to its true value than is achievable by other algorithms. For our task, we can see that the most appropriate method of data imputation is the iterated PCA algorithm.

4.4 Modelling Shape and Structure

In this section, we describe how data imputation using an iterated PCA scheme has been utilised to produce a system capable of modelling structures containing intermittent features. Our models consist of a combined model of the structure of missing data together with a modified PDM. We have called them Structured Point Distribution Models (SPDMs). The overall modelling scheme is represented in figure 4.2. There are two separate phases: shape and structure modelling, followed by a combination step. The following sections describe each in turn.
4.4.1 Shape

For simplicity, figure 4.2 neglects some of the details involved in building a PDM of landmark data. Specifically, no shape alignment is included in the diagram. The data imputation step should be completed after shape alignment. Only after alignment do meaningful relationships exists between landmark coordinates between examples. This section contains details of the various issues that must be addressed when modelling intermittently present features.

Firstly, missing features must be represented within the data vector to be modelled. A missing landmark coordinate is represented by a NaN (the computer representation for Not a Number) in the data vector. For example, shape $i$ with landmark data $[(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)]$ where $(x_3, y_3)$ is unobserved, is represented as the data vector:

$$\mathbf{x}_i = (x_1, x_2, NaN, x_4, y_1, y_2, NaN, y_4)^T \quad (4.9)$$

The first required change is to the alignment algorithm. The standard algorithm aligns all training examples to the mean shape of the set. However, in this case, some of the training data associated with absent features is missing, so a standard mean operation cannot be performed. Instead, we compute the mean value over the available variables:

$$\mu_j' = \frac{\sum_{i} x_{i,P_i}}{n_{P_i}} \quad (4.10)$$

where $P_i$ represents the set of observed variables in example $i$. All shapes are aligned to the shape calculated using equation 4.10.

When aligning shapes $i$ and $j$, only the set of points equal to the intersection of the sets $P_i$ and $P_j$ is used. With these two modifications, alignment proceeds using the
algorithm described in 3.2.

After alignment, shape \( i \) is represented by the data vector:

\[
x'_i = (x'_1, x'_2, NaN, x'_4, y'_1, y'_2, NaN, y'_4)^T
\] (4.11)

where \( x'_i \) represents \( x_i \) after alignment.

To allow a PCA to be carried out, the data vector must not contain missing variables. We must therefore impute the missing values using one of the algorithms detailed in section 4.3. Imputation by any method results in a data vector for shape \( i \) of:

\[
\hat{x}'_i = (x'_1, x'_2, \hat{x}_3, x'_4, y'_1, y'_2, \hat{y}_3, y'_4)^T
\] (4.12)

where \( \hat{x}_3 \) is the imputed value for missing value \( x'_3 \) and \( \hat{y}_3 \) is the imputed value for missing value \( y'_3 \). Alignment does not use imputed values. This has been done to reduce amount of the algorithm that is dependent on imputed data values. Standard PCA can now be applied to the completed training data matrix \( \hat{x}' \) to give a matrix of shape parameters \( b_s \). It is necessary to include information about the presence or absence of landmarks in any instance of the model. Therefore we need to include a method of modelling the structure of the missing data.

4.4.2 Missing Data Structure

Data imputation completes the shape training data and allows a standard statistical analysis to be carried out. However after imputation, the structural information about which values were missing and which were present in the original data is lost. This missing data structure must be modelled to allow correct reconstruction. For this reason, a structural model is built. The structure of the missing data is represented
as a binary vector. In this vector, 1s represent observed data and 0s represent missing data. Shape $i$ has a structural data vector as follows:

$$x_i^s = (1, 1, 0, 1)^T$$ (4.13)

For most applications, including representing the lumen in a model nerve capillary structure, this representation contains a lot of redundant information. However, we have noted in section 4.2 that we wish to retain as much generality as possible in describing missing structures. A less redundant representation would curtail this generality.

We have chosen to use PCA to reduce data dimensionality. PCA of the structure vectors results in a vectors of $n_d$ structure parameters $b_d$, where $n_d$ is the number of principle components retained.

The modes of a PCA of the structural data vectors represent the relationships between the structures in the landmark data. In the case of the nerve capillary landmark data, retaining all positive, non-zero components results in only one mode containing all the variance in the structure vectors. At one extreme, this mode represents the lumen being present, and at the other it represents a the lumen being missing (see figure 4.17). This is to be expected as this is the only structural feature of the data. We have therefore reduced our structural data vectors to a structure parameter vector of length one.

PCA is a linear method of analysis and we are using it to model binary data. The structural parameters $b_d$, which are analogous to the shape parameters of a PDM, are continuous and, of course, reconstruct a continuous structural data vector rather than a binary one. To get back to binary data structure vectors we must threshold the reconstructed values. The optimal threshold position between two normal classes with identical standard deviations is represented diagrammatically in figure 4.13. The solution to the position of the optimal threshold problem is given by equation 4.14.
Figure 4.13: Optimal threshold position between two normally distributed classes.

\[ t = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln \left( \frac{Pr(\omega_2)}{Pr(\omega_1)} \right) \]  \hspace{1cm} (4.14)

where \( Pr(\omega_i) \) is the prior probability of class \( i \), \( \sigma \) is the standard deviation of both classes and \( \mu_i \) is the mean of class \( i \). We can reduce this equation by substituting our known means (\( \mu_1 = 1, \mu_2 = 0 \)) to give:

\[ t_i = 0.5 - \sigma^2 ln \left( \frac{p_i}{1 - p_i} \right) \]  \hspace{1cm} (4.15)

where \( p_i \) is the proportion of examples in which landmark \( i \) has been observed. As the data we are modelling is discrete and therefore not normally distributed, it is not possible to calculate \( \sigma \) directly. However, it is of course possible to produce an estimate of the binary structure vectors in the following way:

\[ \mathbf{\tilde{x}}^* = \mathbf{\bar{x}}^* + \mathbf{P}_d \mathbf{b}_d \]  \hspace{1cm} (4.16)

where \( \mathbf{P}_d \) is a matrix of the first \( n_d \) principal components of \( \mathbf{x}^* \) and \( \mathbf{\bar{x}}^* \) is the mean of the structure vectors. Here, \( \mathbf{\tilde{x}}^* \) is a continuous approximation to the binary structure vectors (\( \mathbf{\tilde{x}}^* \approx \mathbf{x}^* \)) that can be used to calculate estimates of \( \sigma \) and \( \mu \) for both classes, and hence the position of the optimal between class threshold. We note that this approach has not been used in the remainder of the work in this thesis, instead, the
value of $\sigma = 0.25$ has been chosen by observation.

The PCA of the structural data matrix $x^s$ results in a matrix of continuous structure parameters $b_d$, which can then be used, together with the shape parameters $b_s$ to build the combined model of shape and structure.

4.4.3 Combined Model

In sections 4.4.2 and 4.4.1 we described how a training example could be represented by the shape and structure parameters $b_s$ and $b_d$, as shown in figure 4.2. The shape and structure parameters are now used to form a combined model of the shape of the data and its structure. This is done in a way that is similar to the combined shape and appearance model produced when training AAMs. For each training example we generate a concatenated vector:

$$b = \begin{pmatrix} b_s \\ W_d b_d \end{pmatrix}$$

(4.17)

where $W_d$ is a weight for the structure parameters, allowing for the difference in units between the structure and shape models. To equalise the influence of shape and structure parameters this weighting is set to the ratio of total shape and structure variance:

$$W_d = \frac{\sum_{i=1}^{n_s} \lambda_i^s}{\sum_{i=1}^{n_d} \lambda_i^d}$$

(4.18)

where $\lambda_i^s$ is the $i$th eigenvalue retained in the analysis of shape, and $\lambda_i^d$ is the $i$th eigenvalue retained in the analysis of missing data structure. There are $n_s$ shape modes and $n_d$ structure modes.

We now apply PCA to generate a combined model of the data and its structure:
\[ b \approx Qc \]  

(4.19)

where \( Q \) is a matrix of \( t \) eigenvectors of the covariance matrix of \( b \) corresponding to the largest eigenvalues and \( c \) is a vector of combined model parameters which controls both the shape and structure of the data. As \( Q \) is orthonormal, we can obtain \( b \) from \( c \):

\[ c = Q^T b \]  

(4.20)

Therefore, given a vector of combined model parameters \( c \), we can produce corresponding shape and structural parameter vectors (\( b_s \) and \( b_d \)), by first calculating \( b \) and reversing the weighting and concatenation procedure. From these vectors we can reconstruct shape and the structure of any missing data. We now have all the mechanisms required for a generative model.

### 4.4.4 Calculating Model Parameters

Given a list of landmark points we often wish to calculate the model parameters that best represent the data. Section 3.2.1 describes how this is done for PDMs. For SPDMs, calculating combined parameters \( c \) from landmark data \((x, y)\) is a three stage process. Because intermittent features are being modelled, the landmark data \((x, y)\) can contain arbitrary patterns of missing data.

The first stage is to create a binary structure vector, labelling the missing data. This structure vector can be used to calculate structure parameters \( b_d \) in the usual way.

Shape parameters can only be calculated from complete data. Any missing landmark values must first be imputed to complete the data set. Firstly, the landmarks are aligned to the model mean. This is done using only the subset of landmarks present
in both the model mean and data vector. Missing values in the aligned data can now be imputed. In this case, the prior model of the data distribution is known and fixed. The prior model is represented by the modes of variation of the shape model. To impute missing values the following process is carried out:

1. Impute mean values for all missing data.
2. Repeat until convergence of missing values:
   (a) Calculate shape parameters $b_s$ represent the completed data (equation 3.5)
   (b) Reconstruct landmark data from $b_s$ (equation 3.4)
   (c) Update missing data values with new estimates.

Using this algorithm, the best fitting shape parameters $b_s$ are iteratively evolved. We use a convergence criterion that stops iteration when the maximum change in any missing value is less than $10^{-6}$. Note that this imputation scheme can be used to impute missing value estimates for any arbitrary pattern of missing data. It is not restricted to patterns of missing data that have been observed in the training set.

Finally, a weighted concatenation of $W_d b_d$ with $b_s$ (equation 4.17) gives us $b$. Using equation 4.20, combined model parameters $c$ can be calculated.

### 4.4.5 Evaluation

Including extra sources of information in a model of an object should lead to a more specific representation of the object class. In this section, we described an evaluation of the SPDM to determine if including intermittent feature data increases model specificity over a standard PDM that does not represent this information.

Firstly, we use some synthetic data, which enables us to evaluate the SPDM reconstruction against known ‘ground truth’. Two different example applications are
used to evaluate the performance on real data with different types of intermittent appearance:

- capillary objects, containing intermittent LEC boundary data,
- faces, in which moustaches have been annotated where present.

Capillary shape data is highly variable. Therefore, models of capillary shape are only weakly constrained. Including extra shape information should result in an increase in model specificity. It is assumed that excessive constriction of the lumen causes the structure to be practically unobservable. There is therefore a relationship between lumen shape and its presence or absence in each example. A good model should represent this relationship.

Face shape is much more consistent than capillary shape. A model of face shape is strongly constrained by the inter–relationships that exist between the landmark points. The placement of a moustache on an individual is, to a certain extent, arbitrary. The extra constraints introduced by the inclusion of this data will therefore be slight.

**Synthetic Data**

A SPDM was built using synthetic shapes, generated by the following algorithm:

1. Generate a random number $r$, between 10 and 30
2. Plot the points $[(r, 50), (50, 90), (100, 50), (50, 10)]$ forming a kite
3. If $(r < 25)$
   1. (a) draw a square, centred about coordinates $(50, 50)$, with side length $50 - 2r$
4. Otherwise
Chapter 4. Modelling Intermittent Features: Structured Point Distribution and Structured Appearance Models

Figure 4.14: Examples from the set of synthetic training data

(a) put 4 NaN values in the data vector

This algorithm generates structures containing squares within kites, where the first coordinate of the kite and the size of the square are related linearly. When the size of the inner square is less than 0, the square is not plotted i.e. the feature is not present in the image. Due to the random number $r$, the ratio of shapes with a missing feature against those with complete data will be 1:5. Figure 4.14 shows some example shapes from the training data set.

The SPDM was built using 50 training examples, including 99.5% of total shape and structure variation. The model has only one mode of variation, shown in Figure 4.15, which captures the linear relationship between the size of the square and the shape of the kite. We know this is the only correlation in the data due to its synthetic nature. Note how the threshold point of the intermittent feature has been set, using equation 4.14, at the correct position in the mode to remove the square before its side length becomes negative.
Chapter 4. Modelling Intermittent Features: Structured Point Distribution and Structured Appearance Models

Figure 4.15: Synthetic data model: single mode of variation ±2 std. dev.

Nerve Capillaries

A SPDM was built using data from the nerve capillary set. Several images in the dataset contain lumen structures that have become so constricted that they are practically undetectable. Expert annotations of these images vary in whether the structure has been marked at all, leading to an incomplete dataset. A subset of 38 capillaries were selected, 23 of which consisted of annotations of both the basement membrane/endothelial cell material (BMEC) boundary and the lumen/endothelial cell material (LEC) boundary. The remaining 15 annotations contained only the BMEC boundary. Figure 4.16 shows some examples from the training dataset.

The SPDM included 99.5% of the total variance found in the training set, which produced 19 modes of combined shape and structure variation, the first six of which are shown in Figure 4.17. The first mode of the model represents a relationship between BMEC and LEC boundary shape and the presence or absence of the lumen. From right to left across the mode, the lumen appears to become constricted, especially in the horizontal direction. At the far left of the mode, the lumen is modelled as being absent. The model appears to have captured the important relationship between the constriction of the lumen and its absence. The next two modes represent shape relationships, not related to changes in data structure. Weaker shape and structural relationships are represented in modes four, five and six.
Including LEC boundary data in a model of capillaries should increase the specificity of the model. A specific model should only represent valid examples of the object class it represents. Fitting a model to invalid data will result in a reconstruction of the nearest valid shape. Invalid shapes can be generated using random offsets of a valid examples landmarks. Figure 4.18 shows an example of a valid capillary annotation shape (figure 4.18(a)) and an invalid shape (figure 4.18(b)) generated by randomly offsetting each valid landmark position by up to 10% of the size of the capillary in each dimension.

When fitting a model to this invalid shape data, the reconstruction should be constrained to lie close to the original valid shape. Figures 4.18(c) and 4.18(d) show PDM and SPDM reconstructions of the invalid shape. The PDM was built using only the BMEC boundary landmarks without the intermittent LEC boundary. The SPDM was constructed with both boundaries. Both reconstructions have constrained the invalid shape to lie closer to the original valid shape data. Over a set of reconstructions, a more specific model will, on average, produce reconstructions that lie...
Figure 4.17: First six modes of variation from the diabetic nerve capillary model, all $\pm 2$ std. dev. from the mode mean.
Figure 4.18: Generating invalid shapes from a valid example. (a) shows a valid example of capillary shape. (b) is the shape that results after randomly offsetting each landmark of (a) by up to 10% of the size of the capillary in each dimension. (c) shows the PDM reconstruction of the invalid shape overlayed on the valid shape, (d) shows the same figure for a SPDM reconstruction.
closer to the original shape than a less specific model.

We have performed a series of experiments comparing the specificity of PDMs built with BMEC boundary data to SPDMs built with BMEC and LEC boundary data. A PDM and a SPDM were built using the 38 training examples selected above. The PDM contained only landmark data from the BMEC structure, the SPDM was built using all landmarks. Both models contained 99.5% of the total variance within the training set. Each shape in the training set was randomly perturbed to produce an invalid shape. Both models were used to reconstruct the invalid shape. The mean point-to-point distance between reconstructed BMEC landmarks and the original annotation was recorded. This process was repeated varying the maximum size of the random offset between 1% and 25% of the size of the capillary. Figure 4.19 shows examples of PDM and SPDM reconstructions of invalid shape data generated using various maximum offset sizes.

Figures 4.20 and 4.21 show mean point-to-point landmark distances model reconstructions to valid and invalid shape data, plotted against increasing maximum random offset size. The SPDM reconstructions of invalid shapes consistently lie closer to valid shape data than the PDM reconstructions. This is observed at all random offset sizes. Also, PDM reconstructions lie closer to invalid data than SPDM reconstructions. Both these observations indicate that the inclusion of lumen data has increased the specificity of the nerve capillary model. The SPDM will therefore increase the constraints that can be imposed by a model during image interpretation. In chapter 6, it will be shown that using the more constrained SPDM to represent capillary shape and structure leads directly to an increase in capillary segmentation performance.

**Faces**

Intermittent features occur in many structures. Another example is face recognition. For an application, it may be desirable to mark certain areas as facial features that
Figure 4.19: Examples from the evaluation of specificity of nerve capillary models. The solid black line shows the original valid training example, the dotted black line shows the randomly perturbed invalid shape. The maximum size of the random offset is given under the corresponding figure as a percentage of capillary size. The solid red line shows the SPDM reconstruction of the randomly perturbed data, and the solid blue line shows the corresponding reconstruction for the PDM. The SPDM reconstructs shapes that are closer to the original valid shape as the maximum size of random offset increases.
Figure 4.20: Evaluation of the specificity of nerve capillary models, showing the average point-to-point distance of model reconstructions of invalid data to valid shape. The red line shows the distance between the SPDM reconstruction of invalid data and the corresponding valid shape, with error bars of ±1 std. dev. The blue line shows the distance between PDM reconstruction and valid data, also with ±1 std. dev. error bars. SPDM reconstructions of invalid shapes consistently lie closer to valid shape data than PDM reconstructions.
Figure 4.21: Evaluation of the specificity of nerve capillary models, showing the average point-to-point distance of model reconstructions of invalid data. The red line shows distance between SPDM reconstruction and the randomly perturbed, invalid shape data. The blue line shows distance between PDM reconstruction and invalid shape data. Both curves show error bars of ±1 std. dev. PDM reconstructions of invalid shapes consistently lie closer to invalid shape data than SPDM reconstructions.
may not be present in every individual, for example moustaches. A SPDM model was built using 29 face images. The images were annotated by hand with 33 landmarks, marking out the face outline, eyes, nostrils, mouth and moustache if present. Of the 29 images, 9 contained moustaches and the remaining 20 did not. Examples from the training dataset are shown in figure 4.22. The SPDM contained 99.5% of total variance, giving 20 modes of variation, the first six of which are shown in figure 4.23.

For this data, structural information is not involved in the four most significant modes. The shape of moustaches is not obviously related in any structured way to the shape of the rest of the face. Face shape does vary in a highly correlated manner. There is therefore, more significance to the covariance amongst standard face landmarks, and less related to moustache data. For this reason, we see moustache landmarks being represented in only the less significant modes.

We have analysed the specificity of the face model in the same manner as the nerve capillary model. As before, the SPDM was compared against a PDM built with no
Chapter 4. Modelling Intermittent Features: Structured Point Distribution and Structured Appearance Models

Figure 4.23: First six modes of variation from the face model, all $\pm 2$ std. dev. from the mode mean
moustache landmark data. Examples for different sizes of random offset from the evaluation are shown in figure 4.24 and results are shown in figures 4.25 and 4.26.

Figure 4.24: Examples from the evaluation of specificity of face models. The solid black line shows the original valid training example, the dotted black line shows the randomly perturbed invalid shape. The maximum size of the random offset is given under the corresponding figure as a percentage of face size. The solid red line shows the SPDM reconstruction of the randomly perturbed data, and the solid blue line shows the corresponding reconstruction for the PDM. The SPDM reconstructs shapes that are closer to the original valid shape as the maximum size of random offset increases.

Figures 4.25 and 4.26 show that fit of the SPDM to the original data is again closer than the fit of the PDM. The difference is not as large as found in the evaluation of nerve capillary models. This is because the face model is already more highly
Figure 4.25: Evaluation of the specificity of face models, showing the average point-to-point distance of model reconstructions of invalid data to valid shape. The red line shows the distance between the SPDM reconstruction of invalid data and the corresponding valid shape, with error bars of ±1 std. dev. The blue line shows the distance between PDM reconstruction and valid data, also with ±1 std. dev. error bars. SPDM reconstructions of invalid shapes consistently lie closer to valid shape data than PDM reconstructions.
Figure 4.26: Evaluation of the specificity of face models, showing the average point–to–point distance of model reconstructions of invalid data. The red line shows distance between SPDM reconstruction and the randomly perturbed, invalid shape data. The blue line shows distance between PDM reconstruction and invalid shape data. PDM reconstructions of invalid shapes consistently lie closer to invalid shape data than SPDM reconstructions.
constrained than the nerve capillary model, so adding extra constraints to the shape has less of an effect than with the weak capillary model. We also note that, in this case, the PDM model matches approximately as well to the invalid shape data as to the valid annotations. This is somewhat unexpected as the model represents strongly related data and should therefore constrain random data back to valid shapes. Further investigation into the modelling algorithm has revealed that the alignment of the model to the invalid shapes frequently introduces model orientation errors, which are not measured. This change in the orientation of the model can be seen in figure 4.24(d). The comparative increase in specificity of the SPDM over the PDM that we see here could be partially due to the extra constraints introduced by extra landmark data in the pose estimation phase.

At low random offsets, we also note that the SPDM model does not represent the annotation as well as the PDM (figure 4.25. The SPDM contains 99.5% of the total variance of all the landmarks including eight extra moustache landmarks. The PDM contains 99.5% of the variance of the 25 face landmarks alone. Therefore the SPDM must represent a lesser proportion of the variance associated with the face outline data than the PDM. This causes a decrease in the quality of the model reconstruction of annotations and therefore, a slight increase in the distances measured when the model is initialised very close to the annotation. However, this effect is greatly outweighed by the subsequent increase in model specificity when the invalid data is reconstructed.

4.5 Modelling Appearance

In chapter 3 we described Appearance Models, in which shape and grey-level appearance were modelled statistically to improve object recognition. Here we extend this method to include descriptions of the appearance of intermittent features using a structured model with data imputation. We call these models Structured Appearance Models.
4.5.1 Appearance

Appearance can be modelled in a manner analogous to shape (section 3.6). A PCA of a shape–free version of the original images can be performed to produce a set of orthogonal modes of variation. Linear combinations of these modes represent the shape–free appearance of training set and other valid examples in the same way as shape is represented by shape modes. We can use the same process to model the appearance of missing data as we have used to model the landmark positions. As before, this is done by constructing an incomplete training matrix and estimating the missing values using iterated PCA imputation. However, the construction of the data matrix is not as straightforward as it may seem.

The full shape and appearance modelling scheme is shown in figure 4.27. The top two paths through the diagram are equivalent to a SPDM representing and structure information. The bottom path shows the construction of a shape–free appearance representation. This representation contains missing appearance data, so imputation is applied. PCA is applied to the completed data and the resulting appearance parameters are combined with shape and structure to form a combined model.
Shape Free Data Representation

In the case of modelling shape, missing landmark positions were simply represented as missing values in the data matrix. The appearance of intermittent feature that is missing from a particular example can be represented as missing data. However, intermittent feature landmarks that are present may also cause areas of data available in some training examples to be unavailable. To illustrate this, let us take the example of face images marked with an intermittent moustache (section 4.4.5). Figure 4.28 shows a possible shape-free representation for the model together with some examples of warped sample positions. Taking figure 4.28 (a) and (b) together, they suggest that we could represent each training example vector with \( n \) blue samples as follows:

\[
x_a = (f_1, ..., f_n)
\]  

(4.21)

where \( f_i \) is value of sample \( i \). This is the representation used in standard appearance models. However, we know that occasionally a moustache structure is present so we must model its appearance. We can amend the data structure to include an \( m \) sample moustache appearance vector, represented by red circles in figure 4.28(d), giving:

\[
x_a = (f_1, ..., f_n, m_1, ..., m_m)
\]  

(4.22)

where \( m_i \) is the value of sample \( i \) from the moustache. In the case of figure 4.28(b), the values of \( m_i \) for all \( i \) are \( NaN \) as this data is not available. For figure 4.28(d), the moustache appearance values are available. However in this case, we can see that some of the blue samples are mapped to the moustache structure. In fact there is a subset of samples that is repeated within the vector:

\[
(f_k, ..., f_{k+m}) = (m_1, ..., m_m)
\]  

(4.23)
If this data is repeated in this way, we end up with a model where the first $n$ samples from each appearance vector are the same as the samples of a standard appearance model. The patch of data $f_k, \ldots, f_{k+m}$ is being used to represent the appearance of the moustache structure although, in other examples it represents what is present at that position in the data vector if a moustache is not available. To form a good model of appearance, the data vector must be consistent concerning what each element represents. If corresponding variables in the appearance data vectors represent different appearance information, any future analysis is meaningless. Therefore when a moustache is present, this underlying part of the data vector (around the upper lip) must be marked as unavailable and set to $NaN$ values.

The representation can be summarised as follows:

Shape–free appearance: $(f_1, \ldots, f_{k-1}, [f_k, \ldots, f_{k+m}], f_{k+m+1}, \ldots, f_n, [m_1, \ldots, m_m])$

$[f_k, \ldots, f_{k+m}]$ and $[m_1, \ldots, m_m]$ represent the same region of the face. When the moustache is absent: $[m_1, \ldots, m_m] = NaN$. When the moustache is present: $[f_k, \ldots, f_{k+m}] = NaN$.

In summary, care must be taken to consistently assign sample values to the correct part of each appearance vector. It is not sufficient to append data to the end of the appearance vector to represent the appearance of an intermittent features. The corresponding underlying samples must be marked as unavailable when the feature is present.

### 4.5.2 Combined Model

As shown in figure 4.27, once a matrix of shape–free appearance data has been generated, missing data is imputed and a PCA carried out. This results in a set of appearance parameters $b_g$ for each training example. Appearance parameters are combined with shape and structure parameters as follows:
Figure 4.28: Shape–Free representations of face appearance data. Blue crosses and red circles represent sample points. (a) shows sample points in the shape–free representation of face data with no moustache, (b) shows corresponding sample positions after warping to a face shape in the training set with no moustache marked. (c) shows sample points in the shape-free representation with points within the moustache structure circled in red, (d) shows the corresponding warped sample positions in a face shape in the training set.
where \( b_s, b_d, b_g \) are shape, structure and appearance parameters respectively, \( W_s \) is a weight for shape parameters, and \( W_d \) is a weight for structure parameters. To equalise the influence of shape, structure and appearance parameters this weights are set to the ratio of total shape and structure variance against total appearance variance as follows:

\[
W_d = \frac{\sum_{i=1}^{n_g} \lambda_i^g}{\sum_{i=1}^{n_d} \lambda_i^d},
\]

(4.25)

\[
W_s = \frac{\sum_{i=1}^{n_g} \lambda_i^g}{\sum_{i=1}^{n_s} \lambda_i^s},
\]

(4.26)

where \( n_s, n_d, n_g \) are the numbers of model modes representing shape, structure and appearance, and \( \lambda_i^s, \lambda_i^d, \lambda_i^g \) are the \( i \)th largest eigenvalues from a PCA of shape, structure and appearance data respectively.

These weights can also be calculated empirically by varying shape and structure parameters to determine the corresponding magnitude of changes in appearance (see section 3.6).

### 4.5.3 Evaluation

This section describes a qualitative comparison of the structured appearance models against standard appearance models. A good model should only represent valid examples of the object class represented by its training set. Intermittent features cause standard appearance models to represent appearance that is not observed in the
training set. Structured appearance models give a method of modelling intermittent features without affecting the rest of the model.

Structured appearance models of face and nerve capillary images have been built. A qualitative comparison of the model modes produced by structured and standard appearance models is made.

**Faces**

A structured appearance model was built using the same set of images as in the previous section. The images were annotated by hand with 33 landmarks, marking out the face outline, eyes, nostrils, mouth and moustache if present. Of the 29 images, 9 contain moustaches and the remaining 20 do not, examples from the training dataset are shown in figure 4.29. The structured appearance model contained 99.5% of total variance, giving 28 modes of variation, the first six of which are shown in figure 4.30.
Figure 4.30: First six modes of variation from the face structured appearance model, all ± 2 std. dev. from the mode mean
We have also constructed a standard appearance model, from the same data excluding moustache landmark data. This model also contained 99.5% of total variance, represented by 28 modes, the first six of which are shown in figure 4.31.

We can see that the first mode of both models primarily represents skin tone differences present in the training set. However, the standard appearance model contains an indication of a moustache which is not present in the structured model. The centre image in each mode represents the mean of the model. We can see that the mean of the standard model contains a darker patch on the top lip. This type of appearance is not evident in any training example, therefore the mean of the standard model rep-
represents invalid data. The linearity of the PCA means that this appearance distortion is inevitable. Moustaches are represented in the appearance vector of some examples. In other examples, the corresponding part of the appearance vectors represents skin tone. The average of this part of the vector will clearly have values between moustache and skin tone, even though this appearance is not in the training set. This type of distortion is not present in the structured model, and the whole first mode remains unaffected.

Differences can be seen in the second mode of each model. The standard model shows an appearance relationship representing the presence or absence of a dark moustache. Because of the linear nature of the model, a dark area at one end of a mode means there must be a light area at the other. This light area is also not in the training set, and clearly represents invalid appearance. In section 4.4.5 we argued that the presence or absence of a moustache is not particularly significant compared to other shape relationships. The standard appearance model of the data however, represents the relationship as the second most important trend in the data.

The corresponding second mode in the structured model represents mostly shape and appearance change. Other less significant modes of the structured model represent the presence or absence of the moustache feature, in the same way as in section 4.4.5. The moustache feature is turned on at an appropriate point of a mode to allow a dark moustache to be represented, without requiring a corresponding incorrect light area. The significance of the presence or absence of a moustache is now modelled separately from its effect on the appearance of the model.

The standard appearance model of the face images contains several inadequacies. They have been caused by incorrect correspondence of difference appearance types, i.e. certain parts of the shape–free data represent both moustache appearance and skin tone. Structured appearance models represent moustache and skin tone appearance in separate parts of the data vectors. Therefore, the structured appearance model is a better representation of the training set than the standard model.
Nerve Capillaries

A structured appearance model has been compared to a standard appearance model of capillary appearance. A training set of 24 annotations from 10 different capillary images was used to build each model. Examples from the set of images are shown in figure 4.32. The LEC boundary was annotated in all but 5 of the training examples. We constructed a structured appearance model containing 99.5% of the variance within the training set. The resulting model consists of 24 modes, the first six of which are shown in figure 4.33. A standard appearance model was built, this time using only landmarks from the BMEC boundary. Again, the model contained 99.5% of the variance of the training set, resulting in 24 modes. The first six modes of the standard appearance model are shown in figure 4.34.

Again, we can see differences in the corresponding modes of each model. The most dramatic of these is the apparent existence of a black lumen in the first mode of the standard appearance model. This is clearly not present in the data, and neither
Figure 4.33: The first six modes of variation from the nerve capillary structured appearance model, all ± 2 std. dev. from the mode mean. The LEC boundary is marked when it is present in the model instance.
Chapter 4. Modelling Intermittent Features: Structured Point Distribution and Structured Appearance Models

Figure 4.34: The first six modes of variation from the nerve capillary appearance model, all $\pm 2$ std. dev. from the mode mean.
does it feature in the first mode of the structured model. Instead, the appearance of a lumen structure is directly modelled in the second and third models, without distorting the rest of the data. This indicates that the appearance of the capillary structures is not best modelled by a standard appearance model.

Furthermore, we can see in all the modes that the lumen is much more clearly defined in the structured model than in the standard model, where it is represented by a rather diffuse area. The position of the lumen is also represented in the modes of the structured model, where this has been inadequately represented directly in the linear appearance modes of the standard model.

However, adding the ability to model the position of the lumen, and its rather variable shape has also introduced some less desirable features to the model. The structured model modes exhibit some distortion of the appearance of the basement membrane. Positional and shape changes of the lumen structure when coupled with data warping create these local swirling and stretching effects. Careful choice of landmarking strategy has been used to minimise this effect, but it remains in the model modes. These distortion effects are an inevitable consequence of data warping where large variability is present in the data.

4.6 Search

In this section, we describe the extensions we have made to the Active Shape Model (ASM) search algorithm for use with intermittent features. The resulting algorithm has been coupled with the previously described SPDMs to form an active version called Structured Active Shape Models (SASMs). An evaluation of the SASMs shows they perform as well as standard ASMs when applied to face images. Results on their performance when applied to nerve capillary images are given in chapter 6.

We also briefly discuss what is required to use the AAM search algorithm with struc-
Chapter 4. Modelling Intermittent Features: Structured Point Distribution and Structured Appearance Models

tured appearance models.

4.6.1 Structured Active Shape Models (SASMs)

The ASM search algorithm is based on a profile modelling and matching strategy. A full description of the algorithm is given in chapter 3. We have modified this algorithm for use with data representing intermittently present features.

The basic model fitting algorithm consists of constructing a model of sample values along the normal to the shape boundary for each landmark point, then using these models to find the most likely position for that landmark during search. The sampled data for each image for each landmark is called a profile. The following sections describe changes that have been implemented in both of these phases.

Profile Models

The collection of profile data is done at each landmark point that is available in each image. Profiles from missing landmarks are omitted from the subsequent statistical analysis. We have chosen not to impute the data as our imputation method has been designed to avoid skewing the subsequent PCA, not to provide realistic data values. The profile models are therefore based only on observed data, reducing the likelihood of false matches during the search phase.

Model Fitting

The primary modification to the fitting phase is concerned with using image evidence to determine whether an intermittent feature is present or absent.

The structure of the data may change during model fitting as a consequence of the model containing modes that represent both shape and data structure. However, for
Chapter 4. Modelling Intermittent Features: Structured Point Distribution and Structured Appearance Models

this to occur, a very close coupling between a particular shape and a particular data structure must exist in the modes of the model. Also, when a feature is missing in the current iteration of the fitting process, it can have no influence on the profile match positions, and therefore on the new position of the model.

Given a vector of model parameters $\mathbf{b}$, it is possible to produce landmark positions for every feature. The structural model determines which landmarks are present or absent. If we do not delete values for landmarks of missing features, we can produce a position for every landmark in all cases. This allows all profile models to search local images for all landmarks in every case. We have used this to develop an algorithm that allows image data to influence the structure of the model:

1. Reconstruct model parameters $\mathbf{b}$, producing landmarks $(x, y)$.
2. Search local region for profile model best fit positions $(x_{bf}, y_{bf})$.
3. Weight each profile best fit position by quality of model fit $w_{bf}$.
4. For each intermittent feature:
   (a) If the proportion of feature best fit positions $p$ have $w_{bf} < t_{missing}$ mark each feature landmark in $(x_{bf}, y_{bf})$ as missing.
   (b) If the proportion of feature best fit positions $p$ have $w_{bf} \geq t_{present}$ mark each feature landmark in $(x_{bf}, y_{bf})$ as present.
5. Remove features in $(x_{bf}, y_{bf})$ according to the structure model threshold values, except where a feature has been marked as present in step 4.
6. Calculate new model parameters $\hat{\mathbf{b}}$ for points $(x_{bf}, y_{bf})$.
7. Restrict $\hat{\mathbf{b}}$ to lie within valid shape region, update $\mathbf{b}$ with the values of $\hat{\mathbf{b}}$.
8. Unless maximum iteration reached, return to step 1.

This algorithm has three tunable parameters:
1. $p$: the portion of feature points required to mark a feature as present or absent.

2. $t_{\text{missing}}$: the threshold value for profile model match weights to mark a feature as absent.

3. $t_{\text{present}}$: the threshold value for profile model match weights to mark a feature as present.

These parameters together control which landmark data is used to estimate new model parameters $\hat{b}$. At present, these values must be manually chosen. The algorithm will turn on a feature if a certain proportion of the profile model fits to the image data are good, and will turn off a feature if a certain proportion of the fits are bad. Model parameters are restricted to lie within the valid shape region, so only valid class examples will result from this procedure.

**Evaluation**

In this section, we describe an evaluation of SASM search. We have used the values $p = 0.75$, $t_{\text{missing}} = 0.4$ and $t_{\text{present}} = 0.6$, where values of $w_{bf}$ (section 3.3.3) have been scaled to range from 0 to 1. These values were chosen by observation.

The evaluation used the set of 29 face images described in previous sections. These image exhibit considerably less variability and complexity than the nerve capillary images. The model described in section 4.4.5 has been used to search the images.

The following experiment demonstrates that the model fitting algorithm is capable of using model and image data to find an appropriate data structure. Two images were selected from the training set, one in which a moustache has been annotated and the other where it had not. A search of each image was carried out. The searches were initialised at best fit of the model to the other image’s landmark data, i.e. with the moustache feature incorrectly initialised. The starting positions and positions after model fitting are shown in figure 4.35.
model fitting has been corrected in each case i.e. the moustache is now correctly located where it is present, and removed where it is not.

The robustness of SASM model fitting has been compared to the standard ASM fitting algorithm. We have built an ASM and a SASM for the face data. Both models contained 99.5% of total variance. The SPDM was built with all 33 landmarks. The PDM was built with 28 landmarks, which excludes moustache data.

The experiment was designed as follows. Searches were carried out on each image in the training set using both models. Five iterations of ASM or SASM search were carried out in each case. For each example in the training set, an invalid shape has been generated by adding a random offset to each landmark (see section 4.4.5). This invalid shape was used as a starting position for model fitting for both models. The size of the random offset was varied between 1% and 25% of face size along each dimension. Final model fit positions were evaluated by calculating the mean point-to-point distance between final model position and the annotation of the image. Only the 28 landmarks representing all features except the moustache were included in this calculation.

Figure 4.36 shows the mean distance between model fits and annotation data, averaged over the results of searches for the entire training set. This value is plotted against increasing maximum random offset size for both ASM and SASM searches.

The SASM results are very slightly better than the ASM algorithm. The initialisation of each model search was identical, so the slight increase in performance must be caused by the increase in specificity of the SASM shape model. This increase in specificity was demonstrated in section 4.4.5 (figure 4.25). The difference between the results achieved by the models is slight as the extra constraints imposed by moustache data are only small.
Figure 4.35: SASM fitting results: (a) and (c) show model starting positions in green, (b) and (d) show the corresponding final model fits after 5 iterations.
Figure 4.36: Model fitting robustness comparison. Mean point-to-point distance between model fit position and annotated landmarks for a set of image searches. Model searches were initialised at randomly offset annotation data. The size of the maximum offset was varied between 1% and 25% of total annotation size. ASM search results are shown as a blue line, and SASM are shown in red.
4.6.2 Structured Active Appearance Models (SAAMs)

It is not necessary to modify the Active Appearance Model (AAM) search algorithm (section 3.6) at all for use with intermittent features. The entire algorithm is driven by differences in appearance being associated with changes in model parameters. This lends itself well to controlling which features should be used to model a specific appearance.

Evaluation

In this section, we present an example of SAAM search. We have built an SAAM using the face training data described in section 4.5.3. The model fitting algorithm was trained using 203 displacements of pose and model parameters.

The SAAM model was fitted to two images in the same way as the evaluation of SASMs described in the previous section. The model fitting algorithm was run for 5 iterations. The original images, initialisation positions and model fit positions are shown in figure 4.37. We can see that the model fitting algorithm correctly selects the appropriate features to represent the appearance of the image accurately.

In section 3.7, we showed that AAMs were not a suitable image interpretation technique for use with capillary images. For this reason, we have ended our investigation of SAAMs at this point. There is clear potential to the technique. A future evaluation should compare SAAMs against standard AAMs in terms of model fitting robustness and accuracy.

4.7 Summary

In this chapter we have described models that can be used to represent data containing intermittent features. We have presented a novel method of data imputation
Figure 4.37: SAAM fitting results for two training images: (a) and (d) show original images, (b) and (e) show model initialisation each with the moustache feature incorrectly represented, and (c) and (f) show final model fits after five iterations, showing the correct state of the moustache.
that evaluates well against other techniques in the literature for our purposes. This has been used to develop both point distribution models and appearance models, capable of correctly representing data that is only intermittently observed. We have demonstrated that these models can be used to increase model specificity over what can be achieved with standard models. It has also been demonstrated that the modes of our modes are qualitatively more appropriate than the modes of standard models for our data. We have also developed model fitting algorithms that can be used with these models to provide a method of image search.

Our goal was to develop techniques for the modelling of nerve capillary data and this has been achieved. However, in order to make these models totally generic, there are several areas in which further work may be necessary. Specifically, the issue of the weighting of a binary structural vector data against shape data and appearance data needs further thought. Furthermore, the representation of appearance data is currently restricted to structures with a closed boundary.

The important issue to address, in the context of our goal, is whether we are able to exploit the weak constraints in our data. We have shown that the ability to model intermittent features has increased the specificity of the model, and therefore the constraints on fitting to image data.
Chapter 5

Modelling Grey-Level Texture

Constraints on segmentation can be increased by utilising \textit{a-priori} information about grey-level appearance. The appearance of the various structures in nerve capillary images is highly complex and variable. Most information to delineate these structures is contained in their textural appearance. Byrne [18] developed a segmentation scheme, which used snakes driven by a texture based region similarity measure. He used a linear combination of purpose built texture features and Laws texture filters [82] for his texture measure. In this chapter we give a brief overview of texture discrimination methods in the literature, and develop a scheme to model and discriminate between the textures of the basement membrane and the endothelial cell material. The goal of the work is to increase the constraints on segmentation that can be gained from textural appearance. Specifically, this can be achieved by producing a texture discrimination scheme that outperforms Byrne’s measure.

The performance of a texture discrimination scheme is dependent on the quality of the texture features chosen and the classification scheme applied. Therefore, these are the potential areas where improvements can be made. We present a comparative evaluation of four sets of texture features: manually designed texture features, Laws texture features, wavelets and \textit{optimal discrimination} filters. With these, we evaluate linear, quadratic and mixture–model classification schemes. The various schemes are
compared in terms of both misclassification error rate and model fitting performance.

The results of the evaluation indicate that the performance of Byrne’s texture measure can be improved upon, both in terms of misclassification rate and model fitting performance. The result of the comparison differs, depending on whether the texture measure is segmentation or classification. Perfect classification is not possible for capillary images. Some misclassifications can lead to a structured texture image appearance that results in good model fitting performance. Misclassification rate is therefore not always a guide to model fitting performance. We find that the greatest improvement in segmentation accuracy and robustness can be gained using an optimal discrimination filter with a non-linear classification scheme. In terms of misclassification rate, a wavelet texture analysis scheme gives the best performance. Both schemes outperform Byrne’s texture measure. This indicates that the constraints imposed on segmentation have been improved.

5.1 Texture Discrimination

Textures are described in terms of texture primitives, which are the smallest units from which a texture can be built. There are two main approaches to texture classification: statistical and syntactic [62]. Syntactic, or structural, methods describe texture in terms of a repeating patterns of texture primitives. The patterns are represented by a set of rules called a grammar. Syntactic methods are typically useful for modelling textures where there is much regularity, which is often not the case in many natural textures including those observed in nerve capillary images. For this reason, we have focused on statistical texture discrimination.

Statistical methods use pattern recognition techniques on texture primitives which may be the size of pixels. The pattern recognition approach is to classify instances of a texture in an image into a set of classes. The basic notation of pattern recognition is a multi-dimensional feature vector. The feature vector is a set of measurements designed
to encapsulate the relevant characteristics of a texture into a compact region of feature space. The process is split into two phases: training and classification. In training, labelled samples are used to partition the feature space into regions representing the different textures. This can also be done in an unsupervised manner. Annotated training data is available for this evaluation so a supervised training scheme has been used. The partitions of feature space are based on models of the distributions of points. From the resulting partitions in feature space, unseen data can be classified. The choice of features is crucial to the final discrimination performance of the texture measure. A good correlation between the feature vectors and class membership is required. No complex classification scheme will give good discrimination results when applied to bad features, conversely very good features can be separated by almost any method.

5.1.1 Texture Features

An extremely wide and varied body of methods exists for the calculation of texture features [24][26][75][84][88][105][121][146]. We focus on methods that we have evaluated for use with our images, and related techniques.

Grey-level Properties

A simple method of extracting texture features is to use first and second order statistics of the grey-level values within local image regions. Commonly used measures include the following functions, which are calculated over a window of size $(2M+1) \times (2N+1)$ in image $I$ centred at point $(x, y)$:

- Local Average Luminance:

$$f_{av}(x, y) = \sum_{i=x-M}^{x+M} \sum_{j=y-N}^{y+N} \frac{I(i,j)}{(2M+1) \times (2N+1)}$$

(5.1)
• Energy:
\[ f_e(x, y) = \sum_{i=x-M}^{x+M} \sum_{j=y-N}^{y+N} I^2(i, j) \] (5.2)

• Entropy:
\[ f_{en}(x, y) = \sum_{i=x-M}^{x+M} \sum_{j=y-N}^{y+N} I(i, j) \log_2 I(i, j) \] (5.3)

• Contrast (typically \( \kappa = 2, \lambda = 1 \)):
\[ f_c(x, y) = \sum_{i=x-M}^{x+M} |i - j|^\kappa I^\lambda(i, j) \] (5.4)

• Smoothness (number of pixels within a region that lie within \( \tau \) of central pixel):
\[ f_s(x, y) = \sum_{i=x-M}^{x+M} |I(i, j) - I(x, y)| < \tau \] (5.5)

In his texture measure, Byrne used local average luminance, smoothness and entropy. The local region sizes for each measure was \( 5 \times 5 \) \((M, N = 2)\) pixels, with \( \tau = 3 \) for the smoothness measure (the images had 256 grey-levels).

As well as these measures of raw grey-level statistics, comparison of edge frequencies can also be used in texture discrimination [143]. Virtually any edge detector can be used to extract gradient information. Byrne used a distance dependent gradient measure \( g(d) \) which can be computed for any image \( I \) for variable \( d \):
\[
g(d) = |I(i, j) - I(i + d, j)| + |I(i, j) - I(i - d, j)| + |I(i, j) - I(i, j + d)| + |I(i, j) - I(i, j - d)| \] (5.6)

A distance of \( d = 2 \) pixels was used for this gradient function. Byrne set all parameter values manually by inspection. The local energy (equation 5.2) of each function response was calculated, using a Gaussian smoothed region of \( 15 \times 15 \) pixels, to give a texture feature energy. This texture energy was used to build the feature feature...
vectors.

We have re-implemented Byrne’s grey-level statistics texture measure and carried out a comparative evaluation against various other measures. The evaluation is presented in section 5.2.

Spatial Frequency Features

Texture discrimination can also be performed by analysing the spatial frequencies of texture primitives. One example of this is the autocorrelation model of a texture [134]. Autocorrelation coefficients are calculated by:

\[
C_{ff}(p, q) = \frac{MN}{(M-p)(N-q)} \frac{\sum_{i=1}^{M-p} \sum_{j=1}^{N-q} I(i, j)I(i+p, j+q)}{\sum_{i=1}^{M} \sum_{j=1}^{N} I^2(i, j)} \tag{5.7}
\]

where \(p, q\) is the position difference in the \(i, j\) direction, and \(M, N\) are the image dimensions. The values of \(C_{ff}\) can be calculated in the frequency domain from the image power spectrum of the Fourier transform [21] (see next section).

A number of drawbacks exist with spatial frequency texture description methods. The representation is not invariant to even monotonic transforms of the image grey-levels. Also, the efficiency of a frequency only representation is less than combined spatial/spatial–frequency methods, as shall be seen later.

Fourier Analysis

The Fourier transform can be used to calculate spatial frequencies. If a texture is at all directional or periodic, its power spectrum will contain peaks for corresponding spatial frequencies. Fourier texture features are often defined by diving Fourier space into bins. Radial and angular bins are commonly used [26], and are shown in figure 5.1. If the Fourier transform of the image is \(F\), the radial features are defined by:
Chapter 5. Modelling Grey-Level Texture

Figure 5.1: Partitioning the Fourier power spectrum into bins. Axis labels are vertical horizontal spatial frequencies.

\[ V_{r_1r_2} = \int \int |F(u, v)|^2 dudv \]  
\[ (5.8) \]

with integration limits of:

\[ r_1^2 \leq u^2 + v^2 < r_2^2 \]
\[ 0 \leq u, v < n - 1 \]  
\[ (5.9) \]

And the angular features are defined by:

\[ V_{\theta_1\theta_2} = \int \int |F(u, v)|^2 dudv \]  
\[ (5.10) \]

with integration limits of:

\[ \theta_1 \leq \tan^{-1} \left( \frac{v}{u} \right) \leq \theta_2 \]
\[ 0 \leq u, v < n - 1 \]  
\[ (5.11) \]
Radial features are correlated with texture coarseness, and angular features with directionality.

As Fourier transforms can be calculated by a convex lens [130], Fourier features can be calculated almost instantly with optics. However, it is much more usual to use the discrete Fourier transform for this calculation. Fourier techniques are whole image measures. To associate spatial information with the texture features, the image must be partitioned into several small sub-images. These sub-images may be overlapping or not depending on the nature of the discrimination task. The sub-images are transformed and their texture features calculated. This process is inefficient when compared to other spatial/spatial frequency methods. Classification proceeds on the basis of the resulting vectors. Other image transform functions can be used for texture feature calculation in a similar way. The Hadamard and discrete cosine transform [105] have been used for this purpose.

These image transforms result in features in co-ordinate systems that are related to the spatial frequencies of the original texture image [119]. Many texture analysis techniques are inter-related and we note that autocorrelation and the Fourier power spectrum calculate the same subset of second-order texture statistics [143], with corresponding invariance and efficiency problems.

**Laws Texture Energy**

Laws [84][82] designed a set of handcrafted image filters that were matched intuitively to common texture patterns. The patterns can be described as average value, edges, spots and waves. The filters were called texture energy measures, as their output was subject to a local energy calculation. The five basic Laws filters were derived by convolution of three simple vectors with one another: average \( L_3 = (1, 2, 1) \), first difference \( E_3 = (-1, 0, -1) \) and second difference \( S_3 = (-1, 2, -1) \). The resulting five filters are shown in figure 5.2.
Chapter 5. Modelling Grey-Level Texture

$L_5 = (1, 4, 6, 4, 1)$  
$S_5 = (-1, 0, 2, 2, -1)$  
$E_5 = (-1, -2, 0, 2, 1)$  
$R_5 = (1, -4, 6, 4, 1)$  
$W_5 = (-1, 2, 0, -2, -1)$

Figure 5.2: Laws texture filters $L_5$, $S_5$, $E_5$, $R_5$ and $W_5$

Pairs of these filters can be multiplied to produce a 2-D convolution mask. For example, an edge filter can be produced by convolving the image with first the $E_T^5$ mask, then the $L_5$ mask:

$$
E_T^5 \times L_5 = \begin{bmatrix}
-1 & -4 & -6 & -4 & -1 \\
-2 & -8 & -12 & -8 & -2 \\
0 & 0 & 0 & 0 & 0 \\
2 & 8 & 12 & 8 & 2 \\
1 & 4 & 6 & 4 & 1 \\
\end{bmatrix}
$$

(5.12)

As image convolution equivalent to multiplication in the Fourier domain [5], the resulting set of 25 Laws texture filters can be shown to partition the frequency domain into equally sized bins [114] as shown in figure 5.3. In this way, the Laws method is similar to other spatial frequency analysis techniques, and performs well despite its handcrafted nature. However, carrying out the analysis in the image domain using finite response filters gives a spatial location for each measurement. This is a significant advantage in computation efficiency over alternative transforms such as the Fourier transform, which must be windowed to be useful for discrimination.

Byrne used Laws filters as part of his texture discrimination scheme. He used a weighted combination of the following filters: $L_T^5 \times L_5$, $L_T^5 \times E_5$, $S_T^5 \times E_5$, $S_T^5 \times S_5$, $R_T^5 \times R_5$, $E_T^5 \times R_5$ and $S_T^5 \times R_5$. The weight for each filter was determined by a linear discriminant analysis between samples from the endothelial cell material and basement membrane areas. A region of $15 \times 15$ pixels was used for texture energy calculation.
As the Laws method has been shown to be similar to other spatial frequency methods, but is also spatially localised, we have chosen to include it in our evaluation. We have re-implemented Byrne’s Laws texture energy measure and carried out a comparative evaluation against various other measures. The evaluation is presented in section 5.2.

Gabor Filters

Gabor filters have also been used to partition the frequency domain for texture analysis. Gabor filters are Gaussian shaped band-pass filters. Their choice is justified as they seem similar to models of the early vision system of mammals. The filters are also jointly optimal in resolution in time and frequency [39][56].

The equation of a basic even-symmetric Gabor filter oriented at 0° is as follows:

$$ h(k, l) = e^{-\frac{1}{2}\left(\frac{k^2}{\sigma_k^2} + \frac{l^2}{\sigma_l^2}\right)} \cos(2\pi f_0 k) $$

(5.13)
where $f_0$ is the radial centre frequency, and $\sigma_x^2, \sigma_y^2$ are the standard deviations of the filter in the $x$ and $y$ directions. Other orientations can be calculated by rotating this filter with respect to the coordinate system $(k, l)$.

This function can be tuned to respond to certain parts of the frequency spectrum by adjusting $f_0$. This is a useful ability for texture analysis. Jain and Farrokhnia [75] suggest a bank of Gabor filters with dyadic coverage of the radial spatial frequency range and multiple orientations. They suggested using five radial frequencies ($f_0 = \sqrt{2^{1/4}}, \sqrt{2^{1/2}}, \sqrt{2^{3/4}}, \sqrt{2}, \sqrt{2^{3/2}}$) and four orientations (0°, 45°, 90°, 135°) for images of size 256 x 256 pixels. The resulting split of the frequency domain is shown in figure 5.4.
Wavelets

Wavelets have previously been successfully applied to texture classification \(^*\) [24][88][121][146]. The wavelet transform is similar to the Fourier transform, in that it decomposes a signal into a sum of basis functions. The important difference is that the wavelet functions are localised in both frequency (by scaling) and space (by translation), whereas the Fourier basis functions are localised only in frequency. Wavelets are localised waves, their amplitude drops to zero over space instead of oscillating forever. As they retain the relationship between a frequency occurrence and its spatial location, wavelet transforms are potentially useful for texture analysis.

Wavelet functions are derived from a basis function, or Mother wavelet \(\Phi(x)\). The wavelet basis functions are then calculated in the following way:

\[
\Phi_{(s,l)}(x) = 2^{-s/2}\Phi(2^{-s}x - l) \tag{5.14}
\]

Here, \(s\) is a scale factor, and indicates the width of the wavelet function \((2^{-s})\), and \(l\) is the translation index. The basis functions are selected to be orthonormal, so

\[
\int \Phi_{(s_1,l_1)}\Phi_{(s_2,l_2)} = 0 \text{ if } s_1 \neq s_2 \text{ or } l_1 \neq l_2 \tag{5.15}
\]

and it is possible to represent other functions as linear combinations of \(\Phi_{(s,l)}\).

The discrete version of a wavelet transform can be implemented by iterative filtering with a critically sampled filter bank with rescaling [139]. The process is illustrated in figure 5.5. The filter bank consists of a low-pass (average) filter and a corresponding high-pass (difference) filter. At each iteration, the image is down-sampled (\(\downarrow 2\)) by removing every odd numbered element. In this way, wavelets produce a

\(^*\)Wavelets have been studied widely and are applicable to many other areas, notably data compression and noise reduction. Strang and Nguyen [139], amongst others, present a review of the theory of wavelets and their relationship to filter banks.
multi-resolution representation of an image. This implementation of the transform is extremely efficient.

The standard wavelet transform is a dyadic decomposition. This form of decomposition partitions the frequency domain into sub-bands represented in figure 5.6. Mallat [91] used this structure to extract texture features. However, Chang and Kuo [24] showed that useful texture features are most commonly found in intermediate frequency bands, and that the dyadic decomposition is not optimal for texture analysis. They introduced the wavelet packet transform [81], in which the sub-band decomposition is not restricted to be dyadic. Figure 5.7 gives an example of a three-level non-dyadic sub-band decomposition.

Both the discrete wavelet transform and the discrete wavelet packet transform are implemented by critically sampled filter banks with no data redundancy. Unser [146] showed that the use of an overcomplete wavelet representation, called wavelet frames, improved texture discrimination. Wavelet frames are computed by a filtering process without down-sampling. Improvements in spatial resolution are gained at a cost of increased computational and data storage requirements.

There are many choices of mother wavelet. Some of the most well-known include the Meyer [99], Daubechies [38] (of which the Haar wavelet is a specific example) and biorthogonal wavelets. At present, there is no way of optimally choosing a basis function for a particular task.
Figure 5.6: Frequency domain partitioned by a three–level dyadic wavelet transform.

Figure 5.7: Frequency domain partitioned by a three–level non–dyadic wavelet transform.
Wavelet texture analysis is typically found to outperform traditional texture analysis [24], including Gabor filtering [147]. It is a promising technique to apply to our data. We have implemented a wavelet texture feature extraction scheme using wavelet frames with the three-level frequency decomposition shown in figure 5.7. We have used the biorthogonal wavelet basis function to extract texture features.

**Optimal Texture Discrimination**

A typical texture feature extraction technique, such as the filtering methods described in previous sections can be represented by the flow diagram shown in figure 5.8. The diagram represents the filter bank method of texture analysis. Furthermore, each of the filter banks mentioned so far has been designed more or less irrespective of the texture to be analysed. Despite this, these methods have been used to yield successful results in numerous cases. However, heuristically designed texture analysis techniques imply a large set of texture features. If a scheme can be devised that is optimal, with respect to some discrimination criterion over a set of specific textures, the number of texture features will be reduced and robustness improved.

![Diagram](image_url)

**Figure 5.8:** A dataflow diagram illustrating the filter-bank statistical texture analysis process.
Chapter 5. Modelling Grey-Level Texture

One of the first methods of designing an optimal filter bank for texture feature extraction was the eigenfilter method proposed by Ade [2]. For each texture, a $9 \times 9$ matrix was constructed from its autocorrelation function and the eigenvalues and eigenvectors computed. Each $9 \times 1$ eigenvector corresponds to a $3 \times 3$ filter mask. The images were filtered with masks representing 99% of the total variance within the autocorrelation matrix. This led to a significant number of filters for each texture (5 to 9), and corresponding computational requirements. This approach is optimal with respect to texture representation. However, this does not imply that it is optimal with respect to discrimination.

Bovik et al [14] suggested tuning narrow-band Gabor filters to spectral frequency peaks of textures. A semi-automatic scheme of identifying the peaks was proposed. However, like the eigenfilters, this approach is optimal only with respect to texture representation. Dunn and Higgins [45] tuned their Gabor filters to features that yielded the minimum misclassification error for a pair of textures. The optimal centre frequency for each texture was determined by evaluating a wide range of values using the Fourier transform. The bandwidths $\sigma_x, \sigma_y$ of the filter (equation 5.13) were set manually – values of 2, 4, 8 and 16 were suggested.

Randen proposed a method of generating optimal discrimination filter banks [114] [115][116]. He modelled the filter bank texture analysis process (figure 5.8) with a vector formulation. Filtering a texture $x_i(m,n)$ with a mask $h(m,n)$ is represented as:

$$y_i(m,n) = h(m,n) \ast x_i(m,n) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} h(k,l)x_i(m-k,n-l)$$ (5.16)

where $h(m,n)$ is a $M \times N$ filter. If the following definitions are introduced:
Chapter 5. Modelling Grey-Level Texture

\[ h = \begin{bmatrix} h(0,0) \\ \vdots \\ h(0,N-1) \\ h(1,0) \\ \vdots \\ h(1,N-1) \\ \vdots \\ h(M-1,N-1) \end{bmatrix}, \quad (5.17) \]

i.e. the lexicographically ordering of the rows of \( h(m,n) \), and:

\[ x_i(m,n) = \begin{bmatrix} x_i(m,n) \\ \vdots \\ x_i(m,n-N+1) \\ x_i(m-1,n) \\ \vdots \\ x_i(m-1,n-N+1) \\ \vdots \\ x_i(m-M+1,n-N+1) \end{bmatrix}, \quad (5.18) \]

then the output at pixel \((m,n)\) for texture \( i \) (equation 5.16) can be re-written as:

\[ y_i(m,n) = h^T x_i(m,n) \quad (5.19) \]

Similarly, the local energy operation (see figure 5.8) can be written:

\[ v_i(m,n) = w^T z_i(m,n) = w^T y_i^2(m,n) \quad (5.20) \]
where $w$ is the smoothing filter $w(m, n)$ and $z_i$ is the output of the filtering after the non-linear operation $y^2_i(m, n)$. Using this formulation, assuming that the texture is wide sense stationary and that $\sum_{m,n} w(m, n) = 1$, the expected feature value (the feature mean) is:

$$\mu_{v_i} = E\{v_i(m, n)\} = E\{w(m, n) \ast z_i(m, n)\}$$  
$$= E\{z_i(m, n)\} = E\{y_i(m, n)y_i(m, n)\}$$  
$$= E\{(h^T x_i(m, n))(x_i^T(m, n)h)\}$$  
$$= h^T R_{x_i,x_i} h$$

where

$$R_{x_i,x_i} = E\{x_i(m, n)x_i^T(m, n)\}$$

It is noted that $R_{x_i,x_i}$ is constructed from the autocorrelation function of texture $x_i$. The feature variance can also be calculated, assuming the normality of the distribution.

These measures can be used to design filters that are optimal with respect to a discrimination criterion. Unser [146] suggested a criterion that maximises the distance between the means of two classes:

$$J_U(h) = \frac{(\mu_{v_1} - \mu_{v_2})^2}{\mu_{v_1} \mu_{v_2}}$$

(5.23)

Optimisation of $J_U$ entails the solution of:

$$\frac{\partial J_U(h)}{\partial h} = 0$$

(5.24)

over which, the chain rule for differentiation yields:
Combining the partial differential of equation 5.21,

\[
\frac{\partial J_U(h)}{\partial h} = \frac{\partial J_U(h)}{\partial \mu_{v_1}} \frac{\partial \mu_{v_1}}{\partial h} + \frac{\partial J_U(h)}{\partial \mu_{v_2}} \frac{\partial \mu_{v_2}}{\partial h}
\]

(5.25)

with equations 5.21 and 5.25, results in the eigenvalue equation:

\[
(R_{x^2x^2}^{-1} R_{x_1x_1}) h = \lambda h
\]

(5.27)

where

\[
\lambda = \frac{h^T R_{x_1x_1} h}{h^T R_{x_2x_2} h}
\]

(5.28)

The optimal filter is the eigenvector of \((R_{x_2x_2}^{-1} R_{x_1x_1})\) with the the largest corresponding Unser criterion value \(J_U\).

The Unser criterion yields a large separation of mean feature values, but takes no account of feature variances. The Fisher criterion [55] is designed so that an optimal solution will yield features that not only have a large distance between mean values of classes, but produce small variances:

\[
J_F(h) = \frac{(\mu_{v_1} - \mu_{v_2})^2}{\sigma^2_{v_1} + \sigma^2_{v_2}}
\]

(5.29)

Again, to find the optimal solution the partial derivative of the criterion, \(\partial J_F(h)/\partial h = 0\) must be found. Currently, there is no closed form solution for this equation. Randen [114] used various assumptions to approximate the solution.
to be the same as equation 5.27.

We note the similarity of this solution with the eigenfilters of Ade [2]. Both methods require an eigenanalysis of autocorrelation matrices. The important distinction is the optimal discrimination characteristics of Randen’s filters against the optimal representation of eigenfilters. This is effected by the multiplication of the autocorrelation of one class by the inverse of the other class, in an equation derived from a discrimination criterion.

We have chosen to implement an optimal filtering scheme based on Randen’s techniques, to evaluate the effect of using filters tuned to our specific discrimination task.

**Other Statistical Texture Descriptions**

The field of texture analysis is a wide one, and in this work we have chosen to focus on spatial/spatial frequency filter bank methods. The methods described in previous sections are representative of the recent literature on texture analysis. The literature is very old and many other methods have been proposed. There are several important non–filtering texture analysis techniques that cannot be left out of a review, even though we shall not be utilising them. Some of the most well known are described in the following section.

One of the most common methods for texture representation is the co–occurrence method, introduced by Haralick [63]. In this method, a co-occurrence matrix \( P \), is built from the relative frequencies of grey–level pairs of pixels at certain relative offsets. If the image has \( G \) grey–levels the size of \( P \) will be \( G \times G \). If \( G \) is large, the statistical significance will be poor as few pixels contribute to each matrix element \( p_{ij} \). Conversely, if \( G \) is too low, much of the information in the texture will be lost. A value of \( G = 8 \) has been recommended for image patches of size 32 \( \times \) 32 pixels [107]. Texture features consist of statistical measures from the co-occurrence matrix. Haralick suggested 14 measures, of which only a subset is commonly used. Some of
the most common are defined as follows:

- Energy (Angular Second Moment):
  \[
  ASM = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} p_{ij}^2
  \] (5.30)

- Contrast:
  \[
  Con = \sum_{n=0}^{G-1} N^2 \left\{ \sum_{|i-j|=n} p_{ij} \right\}
  \] (5.31)

- Correlation:
  \[
  Cor = \frac{1}{\sigma_x \sigma_y} \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} ij p_{ij} - \mu_x \mu_y
  \] (5.32)

- Entropy:
  \[
  Ent = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} p_{ij} \log p_{ij}
  \] (5.33)

where \( \mu_x, \mu_y, \sigma_x \) and \( \sigma_y \) are means and standard deviations,

\[
\begin{align*}
\mu_x &= \sum_{i=0}^{G-1} i \sum_{j=0}^{G-1} p_{ij} \\
\mu_y &= \sum_{j=0}^{G-1} j \sum_{i=0}^{G-1} p_{ij} \\
\sigma_x &= \sum_{i=0}^{G-1} (i - \mu_x)^2 \sum_{j=0}^{G-1} p_{ij} \\
\sigma_y &= \sum_{j=0}^{G-1} (j - \mu_y)^2 \sum_{i=0}^{G-1} p_{ij}
\end{align*}
\] (5.34)

Another widely used technique is the calculation of model-based texture features. A model of a texture is assumed and its parameters are estimated from a sub-image. The parameters, and attributes derived from them, are used as texture features. An important example of this technique is the autoregressive model of an image [93]:

\[
x(m, n) = \sum_{(k,l) \in N} \theta(k,l)x(m-k, n-l) + \sigma_x \varepsilon(m, n)
\] (5.35)
where $\mathcal{N}$ is the model neighbourhood, $\theta(m, n)$ are the model parameters, and $\sigma_e(m, n)$ is the model error term. The model parameters vary significantly if the texture is fine, but remain mostly steady in coarse textures. The autoregression model has been shown to have similar performance to second-order statistics of spatial frequency, but at a much greater computational cost [57].

Fractal geometry has also been used to model texture [110], especially for natural textures. This method is based on an estimation of the self-similarity of image regions. Chan [23] describes the application of fractal texture measures to the segmentation of structures in electron micrograph images.

### 5.2 Evaluation

This section presents a comparative evaluation of several of the texture feature schemes described in the previous section. Texture analysis has been conducted using: manually designed grey-level statistics, Laws, wavelet and optimal discrimination filter features. We wish to measure the relative performance of each scheme, with the aim of determining the methodology that maximises constraints on image segmentation. Increasing constraints on a complex segmentation problem will lead to an improvement in accuracy and robustness. Therefore, our goal is to find the method of texture analysis that produces an improvement in segmentation performance.

Together with the various texture features, we have evaluated several classifiers: linear, quadratic and two mixture models. Each classifier has been used to model the texture feature data and produce probabilistic texture images. We have made two comparisons of each for each set of texture images: misclassification rate around the BMEC boundary, and ASM fitting performance. Misclassification is a standard method of evaluating the quality of a texture discrimination scheme. However, patterns of misclassification can lead to good model fitting performance. Therefore, misclassification rate does not necessarily correspond to model fitting performance.
For this reason, we have also evaluated model fitting performance directly.

In the next section, we describe the experimental details of our evaluation, and the data that was used. The implementation details of the various filtering and classification schemes are given in sections 5.2.2 and 5.2.3. Results, presented in section 5.2.4, show that improvements over Byrne’s texture measure can be achieved in both misclassification and segmentation. The most significant improvements in model fitting can be made using a mixture model to classify the texture features generated by optimal discrimination filters. The lowest misclassification rate is achieved using wavelet filtering.

5.2.1 Experimental Design

We have evaluated five sets of texture features, and four classification schemes. Each set of texture features were used to train each classifier in turn, giving 20 classification and model fitting experiments.

The evaluations presented have been carried out on a limited set of data. To produce the most meaningful statistics, it is important to have entirely separate training and test data sets, whilst maximising the amount of data analysed. This has been achieved by performing cross-validation experiments. Each classifier is trained on texture feature data from \( N - 1 \) images, where \( N \) is the total number of images available. The model then classifies data from the image left out of the training set, to produce a texture image. This process is repeated for each image, building up a set of texture images. Misclassification and model fitting statistics are calculated from these texture images.

Misclassification

The two-class misclassification rate is given by:
\[ \varepsilon = q_0 \varepsilon_{g_1} + q_1 \varepsilon_{g_0} \]  

(5.36)

where \( \varepsilon_{g_i} \) is the proportion of samples classified incorrectly for class \( i \), and \( q_i \) is the prior probability of class \( i \). \( \varepsilon_{g_i} \) has been estimated as follows: from equation 5.37 (section 5.2.3), the class discrimination boundary is defined as \( I(x, y) \leq 0 \). Therefore, we can create a binary class image \( C(x, y) \) by thresholding at this value. We define the class inside the BMEC boundary as class 0 and outside as class 1.

Therefore, \( \varepsilon_{g_0} \) is the number of pixels inside the BMEC boundary where \( C(x, y) = 1 \), divided by the total number of pixels within the BMEC boundary. The number of pixels outside the BMEC boundary where \( C(x, y) = 0 \), divided by the total number of pixels outside the BMEC boundary gives us \( \varepsilon_{g_1} \). As we have taken training data only from within a region \( \pm 50 \) pixels from the BMEC boundary, we restrict these counts to pixels within this region. In this way, \( q_0 \) is given by the number of pixels within the BMEC boundary divided by the total number of pixels within the sampling region, and \( q_1 = 1 - q_0 \).

Given an annotation of the BMEC boundary, we can calculate \( \varepsilon \). Each image has multiple annotations, and so multiple values of \( \varepsilon \). We have taken the average of these values \( \bar{\varepsilon} \), as the misclassification rate for each image. The misclassification rate for an entire cross-validation experiment is given as the average value of \( \bar{\varepsilon} \) across all images: \( \mu_\bar{\varepsilon} \).

Model Fitting

An ASM search was carried out on each image generated by each set of texture measures. A separate ASM was built for each image, using annotation and texture profile data from every other image. The models contained 99.5\% of the total shape variation in the training set. Combined statistics of normalised texture image patches of \( 5 \times 40 \) pixels were used to train the profile model of the BMEC boundary. Separate
models of selected profile data represented ideal edge data and locally confusing data. 95% of total profile variance was retained in both models.

Five iterations of ASM fitting were carried out on each image. Model parameters were initialised at the best fit to an annotation of the BMEC boundary. This is the best initialisation possible and has been used to determine the best model fitting performance that can be achieved with each set of texture measures. Profile matches were calculated in an area of ±3 pixels along the tangent to the candidate boundary and ±40 pixels along the normal at each model point. The ASM was matched to the profile model best fit positions using a weighted RANSAC model fitting algorithm. 30 landmarks were repeatedly randomly chosen, and used to estimate model parameters. This continued until a consensus set of greater than 90% of best fit positions had been achieved. If after 20 iterations, a 90% consensus had not been achieved, the model parameters with the largest consensus were taken as the model position.

For full details of ASM construction, optimisation and fitting for use with nerve capillary images, see chapter 3.

Model fit results were evaluated by comparison of the point-to-boundary distance of each set of results. This measure is described in section 3.3.4. Summary statistics from each set of model fitting experiments are presented as follows: \( \mu_{all} \), \( \sigma_{all} \) are mean and std. dev. of all point-to-boundary distances between final model positions and each annotation. \( \mu_{acc} \) is average distance for the best 25% of values. \( \mu_{rob} \) is average distance the worst 25% of distances. \( \mu_{acc} \) provides a measure of the accuracy of the best fitting model positions. This gives an indication of the accuracy achievable where good model profile matches are found. \( \mu_{rob} \) provides a measure of the robustness of the model fits.
Chapter 5. Modelling Grey-Level Texture

Data

The evaluation has been carried out on a subset of 10 randomly selected images. Six examples from this set, together with a BMEC boundary annotation, are shown in figure 5.9. Several of the images appear similar, but atypical appearance and locally confusing regions are also represented within the set. Each image has a set of up to four annotations of the BMEC boundary. In total, there are 39 BMEC annotations available.

5.2.2 Filter Banks

This section describes the various methods of image filtering that we have used to generate texture vectors. Filter output is transformed by a local energy calculation to form texture features (see figure 5.8).

We have chosen to estimate the local texture feature energy by squaring each value in a filter response and convolving the result with a Gaussian filter with a standard deviation of 5 pixels. This is equivalent to equation 5.2, calculated in a local window, with a greater weighting for central pixels. When choosing the Gaussian smoothing filter standard deviation, there is a tradeoff between spatial location and reducing ‘noise’. We have chosen the value of 5 pixels by observation. Byrne used the same region size for his local energy calculation. This processes is carried out on the responses of all the filtering techniques described above before any further statistical analysis is begun.

Grey–Level Statistics

We have re–implemented Byrne’s [18] manually designed texture features (see section 5.1.1). The measures used to form the texture feature vector were: local average intensity, gradient, smoothness and entropy. The local region size for each measure
Figure 5.9: Images used in the texture discrimination evaluation. BMEC boundary structure is marked as a yellow line.
was $5 \times 5$ pixels, with a threshold value of 3 grey–levels for the smoothness measure.

**Laws Texture Filters**

Byrne’s texture measure also included combinations of Laws texture filters. He used a weighted combination of the following filters: $L^T_5 \times L_5, L^T_5 \times E_5, S^T_5 \times E_5, S^T_5 \times S_5, R^T_5 \times R_5, E^T_5 \times R_5$ and $S^T_5 \times R_5$. We have re–implemented this measure.

In Byrne’s study, the weight for each filter was determined by a linear discriminant analysis between samples from the endothelial cell material and basement membrane areas. In our case, the texture features will be used as input for several classifiers described in section 5.2.3. Byrne’s texture measure was a combination of his manually designed texture features and Laws filters. We have used the concatenated outputs from the two sets of measures as input to our classification schemes.

**Wavelets**

Our wavelet texture features are generated with a three-level wavelet packet transform illustrated in figure 5.7. We have used biorthogonal basis wavelets (available in the Matlab Wavelet Toolbox as Bior1.5 and Bior3.5 [1]) in this evaluation. The biorthogonal family of wavelets result in texture features that are invariant to orientation differences. The discrete high–pass and low–pass filters for both biorthogonal wavelets have a length of 5 pixels and are shown in figure 5.10.

As with the Laws texture measure described above, the wavelet features have been concatenated with the grey–level texture features. Byrne manually tailored his texture features to specific helpful characteristics observed in the images. The comparison of a completely heuristic texture description scheme to one that utilises problem specific knowledge would not be a fair one. For this reason, we concatenate the manual texture features to every other set of texture features.
Chapter 5. Modelling Grey-Level Texture

Figure 5.10: Biorthogonal wavelet high-pass and low-pass filters used for texture feature generation.

Optimal Filter Bank

We have constructed three optimal discrimination filters of sizes 3×3, 7×7 and 11×11 using the following scheme. The first stage is to extract training data in the form of image patches the same size as the corresponding filter. We wish to discriminate between textures close to the BMEC boundary in each image, so all sample patches have been taken from within an elliptical region ±50 pixels from the best fitting ellipse to the BMEC boundary annotation (see figure 5.12). 50 patch positions from each class (both inside and outside of the BMEC boundary) were chosen randomly from within this region from each image.

As we wish to develop an orientation independent texture measure, four sample vectors were retrieved from each patch position. The four vectors were sampled from patches aligned at orientations of 0°, 45°, 90° and 135° from the x-axis. Each patch is represented as a lexicographically ordered vector consisting of image values from
patch coordinates. Image values were calculated using bilinear interpolation where
patch coordinates did not lie precisely on a pixel coordinate. The vectors were used
to calculate the expected value of the autocorrelation of the texture, and hence the
optimal filter (see equation 5.27).

The resulting three filters are shown in figure 5.11. The filters were generated from
the set of 10 randomly chosen images.

The features from these filters were concatenated with the manual texture features
before classification was carried out.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{filters.png}
\caption{Optimal discrimination image filters generated from the
textures inside and outside of the BMEC boundary.}
\end{figure}
5.2.3 Classification

We have chosen three classification schemes, designed to be progressively more non-linear. The simplest method is linear discriminant analysis which is commonly extended to quadratic discriminant analysis. We have also used a mixture model to represent each class. Combining Gaussian kernels allows us to have an increasingly non-linear model as we increase the number of kernels. All three methods allow us to generate a posterior probability of class membership, and so we can produce probabilistic texture images as well as discrimination statistics.

As the aim of this evaluation is primarily to determine the quality of the texture features, excessive effort in producing a complex classification scheme to improve performance further is not justified.

Each classifier was trained using texture feature data from one of the filter banks described in section 5.2.2.

Training data was taken from regions of width ±50 pixels around the best fitting ellipse to the BMEC boundary annotation. An example of this region is shown in figure 5.12, which shows 100 sample points, taken at random within the selected region. This region has been chosen to focus the data on the textures around the segmentation boundary. Texture in other regions of the image may differ significantly from what we wish to model. The value of the output of each texture feature for the texture scheme is included in a training vector from each sample point. Where the sample position did not lie exactly on a pixel coordinate, the data values were calculated using bilinear interpolation. Each training vector was assigned a class label based on its position on the inside or outside of the BMEC boundary. Feature vectors were taken in this way for each annotation of every image.

1000 sample positions were chosen at random for each image annotation. This gave a total of 39,000 training vectors. The same sample positions were used throughout the entire evaluation. The evaluation was carried out in a cross-validation scheme,
Figure 5.12: Sampling region and 100 random class samples for an example image. Inside BMEC boundary samples are shown as red crosses, outer samples are shown in green.
so data from test image annotations were removed before classifier training began. Once a classifier was trained, it was used to generate a probabilistic texture image. The texture features at each pixel position in the original image were classified. The intensity value in the texture image is assigned as:

\[ I(x, y) = p_1(x, y) - p_2(x, y) \] (5.37)

where \( p_i(x, y) \) is the membership probability of class \( i \) for image coordinates \((x, y)\).

**Linear and Quadratic Models**

Linear and quadratic classifiers model both classes as single multivariate Gaussian distributions. A linear discriminant analysis assumes both groups have the same cross group variance with different means, giving a linear discrimination boundary. A quadratic analysis models the variance of each class separately, leading to a decision boundary with a quadratic form.

**Mixture Model**

A mixture model represents a class distribution as a sum of Gaussian kernels. As such, it subsumes classifiers from quadratic to nearest neighbour, depending on the number of kernels chosen to represent each class. We have chosen to implement the simplest training scheme, with the number of kernels fixed in advance. Two classifiers have been used in this evaluation, one using four kernels to represent each class and the other using eight kernels per class.

We use \textit{k-means} clustering to partition samples from each class into groups, which are then modelled as Gaussian distributions. Cluster centres are initialised randomly within the training data, and samples are assigned to the cluster with the largest
posterior membership probability. After each pass, cluster statistics are recalculated. This process is iterated with samples being re-assigned to clusters with the largest membership probability until no changes occur in the groupings.

The discrimination boundary is now defined as:

$$\sum_{i=1}^{m} p_i \geq \sum_{i=m+1}^{n} p_i$$  \hspace{1cm} (5.38)

where $p_i$ is the posterior probability of belonging to Gaussian $i$, and the distributions from $1, ..., m$ model the first class and those from $m + 1, ..., n$ represent the second class. For any data vector, if this condition is met then the sample is assigned to the first class, otherwise it is assigned to the second.

### 5.2.4 Results and Analysis

This section presents results for each set of texture features and each classifier. Similar results are achieved both in terms of misclassification and model fitting performance for most of the texture features. However, improvements in performance over Byrne’s texture measure (which used combined manual and Laws texture features) are achieved in both evaluations.

**Misclassification**

Mean misclassification rates $\mu_\varepsilon$ and standard deviations for each classifier and each filter bank are given in table 5.1.

At first glance, there is little difference between the texture measures in terms of misclassification rate. All measures produce a misclassification rate of 12%–18%. This reflects the extremely variable appearance of the nerve capillary images. Areas of image evidence around the BMEC boundary with very similar texture to the endothelial
### Table 5.1: Cross-validation mean misclassification rates for each classifier and each set of texture features for 10 randomly chosen images. The standard deviations of the misclassification rates across the set of images are shown in brackets.
cell material have not been distinguished correctly. The correct classification of these areas is extremely hard, if not impossible, using texture information alone, as there is practically no difference in texture to detect. Therefore, perfect segmentation based on texture is not possible.

In general, the results show that the linear classification scheme gives the best results for each set of texture features. The more complex classifiers are too specific and less able to generalise to the characteristics of the texture feature data. However, mixture model classification gives better average performance than quadratic classifiers in each case. Mixture models give a slight improvement as the kernels represent clusters of specific texture types, commonly edge regions and blob areas. This is clearly shown in figure 5.13 where edge regions exhibit a structured response to the mixture model classifiers. The structure is not present in either linear or quadratic images. Slight improvements in overall classification performance lead from this specific texture modelling.

Overall, the wavelet Bior3.5 with a linear classifier give the best discrimination across the BMEC boundary, with a misclassification rate of 0.1219. This represents a 17.7% improvement over the linear Laws texture analysis scheme, which has a misclassification rate of 0.1481. Figure 5.14 shows corresponding texture images generated using Laws and wavelet Bior3.5 texture features with linear classification.

Model Fitting

A summary of model fitting performance is given in table 5.2. The tables give values for $\mu_{\text{all}}, \sigma_{\text{all}}, \mu_{\text{acc}}$ and $\mu_{\text{rob}}$, where $\mu_{\text{all}}, \sigma_{\text{all}}$ are mean and std. dev. of all point-to-boundary distances between final model positions and each annotation, $\mu_{\text{acc}}$ is average distance for the best 25% of values. $\mu_{\text{rob}}$ is average distance the worst 25% of distances. $\mu_{\text{acc}}$ provides a measure of the accuracy of the best fitting model positions. This gives an indication of the accuracy achievable where good model profile matches are found. $\mu_{\text{rob}}$ provides a measure of the robustness of the model fits.
Figure 5.13: Texture images generated using wavelet texture features with different classifiers.
Figure 5.14: Texture images generated using Laws texture feature and wavelet Bior3.5 texture features with linear classification. (a), (c) and (e) were generated using Laws texture features, (b), (d) and (f) used wavelet texture features.
<table>
<thead>
<tr>
<th>Model</th>
<th>$\mu_{all}$</th>
<th>$\sigma_{all}$</th>
<th>$\mu_{acc}$</th>
<th>$\mu_{rob}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>18.3965</td>
<td>19.0737</td>
<td>1.3959</td>
<td>46.7514</td>
</tr>
<tr>
<td>Quadratic</td>
<td>25.3133</td>
<td>25.5013</td>
<td>2.5560</td>
<td>61.8721</td>
</tr>
<tr>
<td>Mixture (4 kernels)</td>
<td>19.5352</td>
<td>19.8761</td>
<td>1.7354</td>
<td>47.4494</td>
</tr>
<tr>
<td>Mixture (8 kernels)</td>
<td>19.1570</td>
<td>16.3232</td>
<td>1.8277</td>
<td>42.5940</td>
</tr>
</tbody>
</table>

(a) Manual.

<table>
<thead>
<tr>
<th>Model</th>
<th>$\mu_{all}$</th>
<th>$\sigma_{all}$</th>
<th>$\mu_{acc}$</th>
<th>$\mu_{rob}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>21.4351</td>
<td>18.4181</td>
<td>2.3199</td>
<td>47.7356</td>
</tr>
<tr>
<td>Quadratic</td>
<td>30.0168</td>
<td>24.6896</td>
<td>4.2274</td>
<td>65.0764</td>
</tr>
<tr>
<td>Mixture (4 kernels)</td>
<td>22.8390</td>
<td>19.3619</td>
<td>2.9880</td>
<td>50.7186</td>
</tr>
<tr>
<td>Mixture (8 kernels)</td>
<td>22.9194</td>
<td>19.1901</td>
<td>3.2375</td>
<td>50.1516</td>
</tr>
</tbody>
</table>

(b) Laws.

<table>
<thead>
<tr>
<th>Model</th>
<th>$\mu_{all}$</th>
<th>$\sigma_{all}$</th>
<th>$\mu_{acc}$</th>
<th>$\mu_{rob}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>17.3995</td>
<td>19.7704</td>
<td>1.2552</td>
<td>45.3178</td>
</tr>
<tr>
<td>Quadratic</td>
<td>30.2336</td>
<td>27.562</td>
<td>3.9265</td>
<td>70.4680</td>
</tr>
<tr>
<td>Mixture (4 kernels)</td>
<td>17.7541</td>
<td>18.4525</td>
<td>1.6766</td>
<td>44.3455</td>
</tr>
<tr>
<td>Mixture (8 kernels)</td>
<td>20.0225</td>
<td>17.1989</td>
<td>2.8143</td>
<td>44.3671</td>
</tr>
</tbody>
</table>

(c) Bior1.5 wavelet.

<table>
<thead>
<tr>
<th>Model</th>
<th>$\mu_{all}$</th>
<th>$\sigma_{all}$</th>
<th>$\mu_{acc}$</th>
<th>$\mu_{rob}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>16.7516</td>
<td>15.2680</td>
<td>1.8170</td>
<td>38.1620</td>
</tr>
<tr>
<td>Quadratic</td>
<td>29.3729</td>
<td>26.7693</td>
<td>4.4212</td>
<td>66.5646</td>
</tr>
<tr>
<td>Mixture (4 kernels)</td>
<td>22.3823</td>
<td>17.8734</td>
<td>3.7281</td>
<td>47.5892</td>
</tr>
<tr>
<td>Mixture (8 kernels)</td>
<td>21.8381</td>
<td>15.9969</td>
<td>4.1898</td>
<td>44.1427</td>
</tr>
</tbody>
</table>

(d) Bior3.5 wavelet.

<table>
<thead>
<tr>
<th>Model</th>
<th>$\mu_{all}$</th>
<th>$\sigma_{all}$</th>
<th>$\mu_{acc}$</th>
<th>$\mu_{rob}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>19.9865</td>
<td>19.8826</td>
<td>1.7487</td>
<td>48.7942</td>
</tr>
<tr>
<td>Quadratic</td>
<td>26.1721</td>
<td>25.7487</td>
<td>2.3028</td>
<td>63.5775</td>
</tr>
<tr>
<td>Mixture (4 kernels)</td>
<td>16.6263</td>
<td>16.3500</td>
<td>1.2775</td>
<td>40.4383</td>
</tr>
<tr>
<td>Mixture (8 kernels)</td>
<td>14.0813</td>
<td>13.2901</td>
<td>1.2177</td>
<td>33.4116</td>
</tr>
</tbody>
</table>

(e) Optimal discrimination.

**Table 5.2:** Model fitting results for texture measures. (a) shows results for manual texture features, (b) for Laws, (c) for Bior1.5 wavelets, (d) for Bior3.5 wavelets and (e) for optimal discrimination filtering. $\mu_{all}$, $\sigma_{all}$ are mean and std. dev. of all point–to–boundary distances between final model positions and each annotation. $\mu_{acc}$ is average distance for the best 25% of values. $\mu_{rob}$ is average distance the worst 25% of distances.
The performance patterns in the model fitting data do not always match those of misclassification data. For example, adding Laws texture features to manually designed features reduced misclassification from 0.1518 to 0.1481, but the corresponding average model fitting distances (18.3965 and 21.4351) show an increase in model fitting error. The characteristics of the Laws texture images are less useful than those of the manual texture measure for model fitting purposes. Some patterns remain consistent; the quadratic classification scheme again gives consistently worse performance across the whole set of evaluations.

Overall, the wavelet and optimal discrimination texture features lead to similar model fitting performance. Both out perform Laws texture images. The best performance is achieved by images generated with mixture model classification of optimal discrimination features. With a mixture model consisting of eight kernels representing each texture class, $\mu_{all} = 14.0813$ pixels. The robustness value ($\mu_{rob} = 33.4116$) is the lowest recorded. The increase in robustness is attributable to the increase in complexity of the images generated with mixture model classifiers. In these images, the number of misleading image regions distant from the BMEC boundary is reduced. This improves robustness when the models are initialised close to the intended solution. It is not clear that these image characteristics will lead to a good segmentation when the initialisation is not so good. Figure 5.15 shows corresponding texture images generated using Laws texture features with linear classification and optimal discrimination texture features with an eight kernel mixture model classification.

Of the images produces by a linear classifier, again the texture features produced by the Bior3.5 wavelet lead to the best results. Histograms of model fit to annotation distance for linear Laws, linear Bior3.5 and mixture modelled optimal discrimination texture images are given in figures 5.16 to 5.18. The histograms show a shifting of values towards zero compared to the Laws histogram for both Bior3.5 and optimal discrimination. The optimal discrimination histogram contains fewer outlying large values, hence a lower value for $\mu_{rob}$. 

233
Figure 5.15: Texture images generated using Laws texture feature and optimal discrimination texture features. (a),(c) and (e) were generated using Laws texture features with a linear classifier, (b),(d) and (f) used optimal discrimination texture features with a mixture model classifier representing each class with eight kernels.
Figure 5.16: Histogram of model fit to annotation distances for images created using Laws texture features with a linear classifier.

Figure 5.17: Histogram of model fit to annotation distances for images created using biorthogonal wavelet Bior3.5 texture features with a linear classifier.

Figure 5.18: Histogram of model fit to annotation distances for images created using optimal discrimination texture features with a linear classifier.
5.3 Summary

In this chapter we have investigated promising methods of texture analysis. We have compared our results with those achieved by a previous texture analysis scheme using Laws texture energy. Wavelet texture features and those obtained from an optimal discrimination analysis both outperform the Laws texture analysis scheme. The lowest misclassification rate was achieved using features from a biorthogonal wavelet decomposition, using a linear classifier. This texture measure gave a 17.7% improvement over the Laws texture measure. The best model fitting performance was achieved using texture features from optimal discrimination filters. These features were classified with a non-linear mixture model. A wider evaluation is necessary to determine whether this performance is repeatable given different initialisation conditions. In chapter 6, we texture process the entire set of capillary images using both schemes. We segment each set of images using SASM search, and find that wavelet based texture images give the best performance.

The goal of the work presented in this chapter was to strengthen constraints on segmentation. The improvements recorded over a previously published texture analysis scheme are slight, but indicate that the information in the textural appearance of the nerve capillary images is being extracted and utilised more successfully. Therefore extra constraints have been added to segmentation. In common with the optimisations made in chapter 3, the improvement is not overwhelming. The work in this chapter represents another incremental step in achieving improved segmentation results.
Chapter 6

Nerve Capillary Segmentation

This chapter presents an evaluation of nerve capillary segmentation using model driven techniques. In this thesis, a number of developments have been made to improve automatic capillary segmentation: grey–level modelling, robust model parameter estimation, SPDMs and texture discrimination. The benefits of each development have been investigated in chapters 3, 4 and 5. However, the only true measure of any benefits introduced by these techniques is how they perform during image search. The experiments described in this chapter evaluate the segmentation performance that can be achieved. Results are evaluated against expert annotations and compared with Byrne’s data–driven segmentation scheme [18].

The evaluations in this chapter have been design to compare the effect of the following segmentation factors:

- SPDMs compared to PDMs,
- Bior3.5 wavelet compared to optimal discrimination texture images and
- single and multi–resolution search compared to genetic algorithm model fitting.

The extra constraints imposed by the more specific SASM are shown to improve
segmentation performance for the set of capillary images. Wavelet and optimal filtering based texture images are compared, with wavelets producing slightly improved segmentation performance.

Multi-resolution image search is found to be an unsuitable segmentation method. Errors introduced by areas of misleading image evidence at low image resolutions propagate through the search, leading to poor final segmentations. Fully automatic genetic algorithm search has been found to give the most accurate segmentation results. However, this has been achieved with a large increase in computational cost.

Overall, segmentation using model-based search produces results which are more accurate than previous data-driven approaches. The resulting boundaries can compare favourably with manual segmentation. However, the segmentation is insufficiently robust to be used in isolation for routine analysis.

6.1 Experimental Design

All experiments in this chapter use a cross-validation scheme to maximise the significance of the results. The general experimental procedure is as follows:

1. For every capillary image $I$
   
   (a) Create a texture image $I_t$, using texture features derived from every image except $I$

2. For every texture image $I_t$
   
   (a) Build model with annotation and profile data from every image except $I_t$
   
   (b) Fit the model to image $I_t$
   
   (c) Record and evaluate the final model fit position
In chapter 4, we described an extension to PDMs called Structured PDMs. Structured PDMs can represent the intermittently present LEC boundary and were shown to be a more specific model of capillary shape. In this chapter, they are evaluated against PDMs to determine how a more specific model affects segmentation performance. Each experiment consists of ASM and SASM searches of the set of capillary texture images. The searches are initialised in the best possible position. This initialisation has been used to measure the best performance achievable for each model under ideal conditions.

In chapter 5, wavelet texture images were found to produce the smallest misclassification rate in recognising basement membrane and endothelial cell texture. However, images produced using optimal discrimination texture features with a non-linear classification scheme were shown to result in the best model fitting performance. These images were found to be highly complex. It was speculated that this complexity may degrade model fitting performance in the absence of an accurate initialisation. To determine the best texture analysis to use for capillary segmentation, we have compared the model fitting performance of SASMs, initialised at the model mean shape, for both wavelet and optimal discrimination texture images.

The results of local refinement model fitting schemes, such as SASM fitting, are dependent on the quality of initialisation and search space characteristics (see figure 2.1). Byrne used manual initialisations for his snake segmentation scheme. Manually initialising model fitting experiments is a time consuming and subjective process. Multi-resolution image search can be used to reduce the necessity of an accurate initialisation by allowing a larger image region to be efficiently searched. Although this process evaluates a larger image area for each model fit, the technique is still a local refinement of an initial position.

Genetic algorithms provide a more global method of model fit optimisation. They have been previously applied to PDM fitting [68][70] and provide an efficient method of searching large, bounded parameter spaces. We have compared the segmentation
performance of multi-resolution and genetic algorithm model fitting.

In the rest of this section, we describe the following parts of the evaluation scheme in more detail: annotation data, model building, texture image generation and model fitting.

### 6.1.1 Annotations

Each nerve capillary image has up to four associated annotations of the BMEC boundary. Three of the annotations have been provided by clinical experts on separate occasions. The fourth was formed by manual adjustment of Byrne’s snake segmentations by a clinician. Several images were rejected by the clinical experts during annotation on each occasion, leaving a set of 33 images with 131 annotations. The annotations have been smoothed using a Fourier descriptor model [136] to reconstruct the boundary with 10 harmonics. The smoothed annotations have been sub-sampled and consist of 50 points evenly spaced in distance around the BMEC boundary. This data has been used to train shape models and evaluate segmentation performance. Section 3.3.1 describes the landmarking scheme, and justifications for its choice.

There is considerable variation in the position of the BMEC boundary annotation for each image. We have evaluated this variation in terms of the point-to-boundary distance (see section 3.3.4). For each set of associated image annotations, we have calculated the distance between each point on an annotation boundary and the closest different annotation. These distances have been calculated for each annotation of every image. A summary of the point-to-boundary distances is shown in table 6.1. The average distance ($\mu_{all}$) of each point on an annotation to another annotation of the same image is 4.05 pixels. The largest 25% of values ($\mu_{rob}$) have an average distance of 12.2 pixels.

The histogram of annotation to annotation distances is shown in figure 6.1. The tail of the histogram (figure 6.1(b)) shows many medium sized distances between

240
Table 6.1: Summary of point-to-boundary distances for annotations of the same image. $\mu_{all}$, $\sigma_{all}$ are mean and std. dev. of all annotation to annotation distances. $\mu_{acc}$ is mean of the closest 25% of measurements. $\mu_{rob}$ is mean distance for the worst 25%.

<table>
<thead>
<tr>
<th>Annotations</th>
<th>$\mu_{all}$</th>
<th>$\sigma_{all}$</th>
<th>$\mu_{acc}$</th>
<th>$\mu_{rob}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4.0506</td>
<td>8.0642</td>
<td>0.3296</td>
<td>12.1965</td>
</tr>
</tbody>
</table>

Figure 6.1: Histogram of point-to-boundary distances for annotations of the same image. (a) entire histogram, (b) tail of the distribution.
annotations of corresponding images. A few large distances of up to 80–90 pixels are present in the distribution.

These figures and distribution represent considerable variation in the marked BMEC boundary positions. Figure 6.2 shows four examples from the set of annotated capillary images. Each image is labelled with its average annotation to annotation distance ($\mu_{im}$). The most ambiguous set of annotations for a single image are shown in figure 6.2(a). This image shows several completely distinct interpretations of the position of the BMEC boundary. Figure 6.2(b) shows the most consistent set of annotations. In this case the position of the boundary has been drawn with only fine scale differences on each occasion, despite the presence of locally confusing image evidence. Figures 6.2(c) and 6.2(d) are examples where $\mu_{im}$ is similar to the overall average $\mu_{all}$. In both cases, the boundary annotations are in the most part consistent. However, parts of the boundaries have been annotated differently on different occasions.

In many images, the lumen is so constricted that it is practically unobservable. For this reason, annotations of the LEC boundary are not available for each corresponding BMEC boundary annotation. In total, 88 lumen annotations are available for the corresponding 131 BMEC boundaries. In 43 cases, the LEC boundary has not been annotated. LEC boundary annotations have been smoothed in the same way as the BMEC boundaries. Each boundary is represented by 50 evenly spaced points, aligned to the corresponding BMEC boundary annotation. Where necessary, if the LEC is not marked, the 50 points coordinates are filled with the computational representation of Not a Number (NaN). The annotation landmarking process is described in detail in section 3.3.1.

6.1.2 Shape and Structure Modelling

Two modelling schemes have been chosen to represent the shape, structure and appearance variation within the nerve capillary structures. ASMs (chapter 3) have been used to model BMEC boundary shape variation. SASMs (chapter 4) extend the
Figure 6.2: Example annotated BMEC boundaries exhibiting ambiguous interpretations. $\mu_{im}$ is the average point-to-boundary distance between each annotation point and the nearest different annotation boundary for each image. (a) shows the most ambiguous set of annotations, (b) is the most consistent, (c) and (d) are examples with $\mu_{im}$ similar to the overall average $\mu_{all}$. 

(a) $\mu_{im} = 16.2886$  
(b) $\mu_{im} = 1.5666$  
(c) $\mu_{im} = 3.0511$  
(d) $\mu_{im} = 3.9749$
ASM paradigm to allow representation of the intermittently present LEC boundary. Both techniques represent local grey-level appearance around landmarks using profile models. These profile models are used to evaluate image data during model fitting.

**ASM**

ASMs used in this evaluation have been built with 50 landmark points from the BMEC boundary. The model built to segment a specific image utilises all annotation data except that from the image to be segmented. The shape models contained 99.5% of the total variance in the training data. Model construction is described in detail in section 3.2.

**SASM**

SASMs have been built with landmarks from both the BMEC and LEC boundaries. Both boundaries were represented by 50 points where present. The models contained 99.5% of variation within this training set and were also constructed using a leave-one-out scheme. SASMs are described in detail in chapter 4.

**Profile Models**

The normal practice in ASM training is for each model point to have an associated profile model. We showed in section 3.3.2 that better results could be obtained by using a single combined profile model for all BMEC model points. This model includes two distinguishable types of boundary appearance – those that exhibit a relatively clean step edge in texture image intensity at the BMEC boundary, and those where confusing regions of image evidence lie close to the landmark (see figure 3.3.2). In these evaluations we model both types of appearance using a mixture model. Each mixture model kernel represents 95% of total training set variance. Image profile
patch dimensions of $40 \times 5$ pixels are used. The profile modelling process is described in detail in section 3.2. The model building parameters were determined from the evaluation presented in section 3.3.5.

SASM profile mixture models are built using the same parameters as ASMs for the BMEC boundary profile models. The profile models for the LEC boundary are built using combined, normalised data from the LEC boundary annotations, represented by a single Gaussian kernel. Profile patch size is $40 \times 5$ pixels and 95% of total variance is included in the model.

6.1.3 Texture Images

Two sets of texture image are evaluated. Input data consists of the set of capillary images together with associated BMEC boundary annotation data. Each image from the original set is processed with Bior3.5 wavelet and optimal discrimination filter banks. In each case, sample vectors are taken from each image on either side of the BMEC boundary and used to train a classifier. For image $I$, vectors from every other image are used for training. The resulting classifier produces the texture image corresponding to image $I$. The texture image generation process is described in full in chapter 5.

Wavelet Texture Analysis

We have used the biorthogonal wavelet Bior3.5 [1] with a two-level wavelet packet decomposition to generate our texture features. The details of the scheme are given in section 5.2.2.

The details of wavelet texture image production are as follows. 1,000 training vectors are taken from each image, from a region within 50 pixels of the BMEC boundary. A linear classifier is trained to discriminate between endothelial cell material and
Chapter 6. Nerve Capillary Segmentation

Figure 6.3: Examples from the set of texture images generated using a linear classification of biorthogonal wavelet texture measures.

the basement membrane for each image. The classifiers are used to produce the probabilistic texture images. Examples from the set of wavelet texture images are shown in figure 6.3.

Optimal Discrimination

Three optimal discrimination filters are generated using the algorithm described in section 5.2.2. The filters have sizes $3 \times 3$, $7 \times 7$ and $11 \times 11$. Each image is filtered with this set to produce texture features. As before, 1,000 samples are taken from each image and used to train classifiers. Classifiers are designed as mixture models representing each texture class with eight Gaussian kernels. A $k$-means cluster analysis is carried out on each texture class of training data. The set of training vectors
Figure 6.4: Examples from the set of texture images generated using a mixture model classification of optimal discrimination texture measures.
defining each cluster is used to define the mean and covariance of a Gaussian kernel.

The mixture model classifiers are used to produce a probabilistic texture image for
each original image. Examples from this image set are shown in figure 6.4.

6.1.4 Model Fitting

Three different model fitting techniques are evaluated. The simplest is a single resolu-
tion local search using ASM and SASM model fitting algorithms. This is extended to
a multi−resolution scheme, again using profile model matching as the basic technique.
The characteristics of the search space of some images cause local refinement tech-
niques to fail. Genetic algorithms have been implemented as a global optimisation
technique to overcome this.

Local Refinement

An extension of ASM fitting was evaluated in section 3.3.5. Outlier tolerant RANSAC
model parameter estimation was incorporated to the basic profile matching scheme
and found to improve model fitting performance.

In the evaluations in this chapter, profile models search regions with size ±40 pixels
along the normal to the candidate boundary position, and ±1 pixel along the tan-
gent. The RANSAC process uses a randomly chosen set of 30 candidate landmarks
to estimate model parameters. If the consensus set exceeds 90% of the total number
of candidate points, the parameters are accepted as the model position. If after 20
iterations, consensus has not been achieved, the model fit with the largest consensus
set is taken as the fit position. These RANSAC control values were chosen by in-
spection. The model fitting process is carried out for 5 iterations. The ASM fitting
technique is described in detail in section 3.2.2, with extensions and optimisations
specific to capillary segmentation described in section 3.3.5.
SASM model fitting uses the same algorithm with three extra parameters to control the presence/absence of the LEC boundary. If 80% of profile model fits for the LEC boundary have a probability of less than 0.25, the LEC is marked as missing. If 80% of profile model fits for the LEC boundary have a probability of greater than 0.5, then the LEC is marked as present. These threshold values were again chosen by inspection. If neither of these conditions are met, the state of the LEC remains as it was in the previous iteration. SASM model fitting is described in section 4.6.1.

Multi–Resolution

Multi–resolution search consists of several single resolution model fits on a Gaussian image pyramid. In this evaluation, three image resolution levels are used. Separate profile models are built for each resolution level. Profile model dimensions remain fixed, so image sub–sampling means the profile model effectively represents a larger area of the original image. At each resolution level, all model fitting search area parameters remain the same as the single resolution case. This means that larger image areas are considered at lower resolutions.

Model fitting begins on the lowest resolution image. A single resolution search is carried out using image profile data from the same resolution level. At each level, the model position is initialised at the best fit of the current model to the candidate boundary of the previous resolution level. Five iterations of model fitting are performed at each level. Multi–resolution ASM model fitting is described in section 3.2.2.

Genetic Algorithms

The ASM and SASM fitting algorithms contain a fundamental limitation that prevents the correction of model orientation. Figure 6.5(a) shows an elliptical model built from four landmark points: \((p_1, p_2, p_3, p_4)\). If the model is incorrectly aligned
Figure 6.5: Fitting an incorrectly oriented shape model to image data.

to image data, as shown, and there is no structural variation in the boundary, local image search will result in best fitting profile model positions \((q_1, q_2, q_3, q_4)\). Point–to–point Procrustes alignment of the model points to the best fit positions will not alter the orientation of the model. If the shape of the model is constrained to be ellipsoid with a limited aspect ratio range, shape estimation will result in the model fit shown in figure 6.5(b). If we make the assumption that model points can be matched to any position on the desired boundary, an alternative boundary–to–boundary alignment process would result in a closer fit of model points to image data.

We have assumed that there is no consistent structural variation in appearance around the BMEC boundary. PDM model points have been aligned with the major axis of the best fitting ellipse to the annotation boundaries. The shape model training set therefore contains data with a limited range of aspect ratios. For these reasons, incorrect orientation will not be corrected by ASM search, even when perfect image best fit positions are available.

There is another limitation of local refinement ASM search. Nerve capillary images contain many regions of locally consistent, but misleading information. These image regions cause local refinement methods to produce inaccurate segmentations, unless extremely accurate initialisations are available. The effect of this characteristic has
been reduced by applying an error tolerant RANSAC model fitting strategy with explicitly modelled local evidence appearance. However, it may be necessary to perform a more global optimisation over model position and parameters.

To address the orientation correction and local image evidence issues, we have implemented a floating point genetic algorithm model fitting algorithm. A description of genetic algorithms is given in section 2.3.2.

Each candidate boundary is represented by a floating point chromosome vector consisting of model parameters and pose information. A crossover probability of 0.5 has been used with a mutation probability of 0.05 for each chromosome element. In each generation, one iteration of local model fitting is carried out for each individual. The search area used for this local refinement is $\pm 15 \times \pm 1$ pixels. The fitness function of each chromosome is defined as the average probability of all the profile model fits in the RANSAC consensus set. Breeding pairs are selected using a roulette selection process that favours chromosomes with high fitness values. The individual with the largest fitness function in a population is retained unchanged through to the next population. A population of 100 individuals is maintained throughout the search. Model fitting is carried out until 20 generations have elapsed. We have used a Matlab implementation of genetic algorithms [73].

The final solution of the genetic algorithm is taken to be the individual in the final population with the highest fitness function.

6.2 Results and Analysis

Results from nerve capillary segmentation experiments are presented in this section. Each set of results has been calculated from segmentations of each capillary image. The comparisons we make are as follows:

ASM vs SASM as a modelling basis. We show that the additional constraints intro-
duced by the SASM improve segmentation results.

Wavelet vs optimal discrimination as a texture representation. We show that, contrary to the findings if section 5.2.4, the additional complexity of the optimal discrimination representation results in a degraded segmentation performance.

Single and multi-resolution hill climbing vs genetic search. The methods are compared on their ability to reach a good segmentation using arbitrarily placed initialisation of the the search. The best performance is achieved using genetic search.

### 6.2.1 ASM vs SASM – Modelling Constraints

This evaluation investigates the effects of increasing model constraints on a complex image segmentation task. SASMs have been shown to be more specific (section 4.4.5) and so potentially impose greater constraints on segmentation than ASMs. Each search was initialised in the best possible position. Model parameters were initialised to the best fit to an image annotation.

<table>
<thead>
<tr>
<th></th>
<th>$\mu_{\text{all}}$</th>
<th>$\sigma_{\text{all}}$</th>
<th>$\mu_{\text{acc}}$</th>
<th>$\mu_{\text{rob}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASM</td>
<td>6.2292</td>
<td>8.8341</td>
<td>0.4946</td>
<td>17.1045</td>
</tr>
<tr>
<td>SASM</td>
<td>5.5998</td>
<td>8.4055</td>
<td>0.4761</td>
<td>15.3063</td>
</tr>
</tbody>
</table>

Table 6.2: Model fit to annotation point-to-boundary distance summary, for ASM and SASM searches. $\mu_{\text{all}}$, $\sigma_{\text{all}}$ are mean and std. dev. of all model fit to annotation distances. $\mu_{\text{acc}}$ is mean model fit to annotation distance of the best 25% of model fit points. $\mu_{\text{rob}}$ is mean distance for the worst 25% of model fit points.

Table 6.2 shows a summary of model fit to nearest annotation distances. The more constrained SASM searches show improvements over the ASM results in every measure. Overall mean error has been improved from ASM $\mu_{\text{all}} = 6.2292$ to SASM $\mu_{\text{all}} = 5.5998$. Accuracy and robustness have been improved and variability reduced. The introduction of the extra constraints imposed by the inclusion of LEC boundary
Figure 6.6: Histogram of ASM fit to annotation point–to–boundary distances. (a) shows the entire distribution, (b) shows only the tail of the histogram.
Figure 6.7: Histogram of SASM fit to annotation point-to-boundary distances. (a) shows the entire distribution, (b) shows only the tail of the histogram.
data, has resulted in measurable model fitting improvements. The differences measured are small, but have been achieved over a set of extremely complex images. The significance of the differences in the results has been tested using Mann–Whitney and two sample Kolmogorov–Smirnov tests [131]. The null hypothesis requires that both sets of results are drawn from the same distribution, and the alternative is that they were drawn from different distributions. Using both tests, the null hypothesis can be rejected at a significance level of 0.01.

Figures 6.6 and 6.7 show histograms of point–to–boundary distances for each model fit position to the nearest annotated boundary. The majority of values in the SASM histogram (figure 6.7) have been shifted toward the origin when compared with the ASM histogram (figure 6.6). Notice that the number of gross outliers is similar in both histograms. This will be discussed in section 6.3.

Figure 6.8 shows example ASM and SASM segmentations, together with annotations of the BMEC boundary. Figures 6.8(a) and 6.8(b) show the most accurate results achieved for each segmentation. Notice that this accuracy has been achieved in the presence of significant amounts of locally consistent but misleading image information around the BMEC boundary.

In figure 6.8(d), the extra constraints imposed by LEC profile models and shape relationships have clearly improved the final SASM fit position. The corresponding ASM segmentation position is shown in figure 6.8(c). Extra constraints are especially useful where image evidence is weak, as the the case in this image.

For both models, failures have occurred even when the initialisation is the best possible. Figures 6.8(e) and 6.8(f) show an example of an image where both ASM and SASM segmentations have failed. We discuss this further in section 6.3.
Figure 6.8: Example segmentation results. Final model fit positions are shown in red, with all BMEC boundary annotations marked as yellow lines. (a), (c) and (e) show ASM search results, (b), (d) and (f) show corresponding SASM results.
6.2.2 Wavelet vs Optimal Discrimination Texture Analysis

We have evaluated the SASM segmentation performance for wavelet and optimal discrimination texture images. Five iterations of single resolution SASM search were carried out on each image in each set. Segmentations were initialised at the mean model shape. The mean shape was aligned randomly within $\pm \pi/4$ of the best fit orientation for an annotation. The scale of the initialisation was set to lie within $\pm 25\%$ of the best fit scale, and the best fit position was offset by up to $\pm 20$ pixels in each direction.

<table>
<thead>
<tr>
<th></th>
<th>$\mu_{all}$</th>
<th>$\sigma_{all}$</th>
<th>$\mu_{acc}$</th>
<th>$\mu_{rob}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biorthogonal wavelet</td>
<td>16.0663</td>
<td>20.4655</td>
<td>1.1613</td>
<td>43.2894</td>
</tr>
<tr>
<td>Optimal discrimination</td>
<td>16.5527</td>
<td>22.8987</td>
<td>1.0405</td>
<td>47.5405</td>
</tr>
</tbody>
</table>

Table 6.3: Model fit to annotation point–to–boundary distance summary, for SASM searches of wavelet and optimal discrimination images.

Table 6.3 shows a summary of model fit to nearest annotation distances. Point–to–boundary distances for each image set are similar. Wavelet images result in slightly better overall mean, variance and robustness results. Optimal discrimination images give a very slightly improved accuracy. Wavelet segmentation results are significantly better than optimal discrimination results, using the Mann–Whitney and two sample Kolmogorov–Smirnoff tests at the 0.01 significance level.

Figures 6.9 and 6.10 show histograms of all point–to–boundary distances for the entire set of wavelet and optimal discrimination texture image searches. There are a greater number of large outlying distances in the optimal discrimination histogram (figure 6.10) than in the wavelet data (figure 6.9). This indicates that optimal discrimination texture images result in a larger number of inaccurate segmentations.

Figure 6.11 shows corresponding examples from the set of wavelet and optimal discrimination segmentations. The accuracy of segmentations is occasionally improved.
Figure 6.9: Histogram of SASM fit to annotation point–to–boundary distances for wavelet texture images. (a) shows the entire distribution, (b) shows only the tail of the histogram.
Figure 6.10: Histogram of SASM fit to annotation point-to-boundary distances for optimal discrimination texture images. (a) shows the entire distribution, (b) shows only the tail of the histogram.
in the optimal discrimination images as is illustrated in figures 6.11(a) and 6.11(b). In this example, the optimal discrimination texture image contains a much clearer delineation at the BMEC boundary than the weaker evidence in the wavelet texture image. Frequently however, the optimal discrimination images contain much complex structure. This structure is not useful in identifying the correct segmentation boundary and degrades model fitting performance. This can be seen in figures 6.11(c) and 6.11(d), where the optimal discrimination segmentation as been perturbed in several places by confusing image evidence.

Gross failures again occur in both image sets for the capillary image that contains an unusually compressed structure (figures 6.11(e) and 6.11(f)).

### 6.2.3 Automatic Segmentation

Results presented in previous sections were achieved using single resolution SASM image search. The SASM fitting method is a local refinement to image data. Multi-resolution ASM fitting allows a larger image region to be searched efficiently. Genetic algorithms are a global optimisation technique.

Single and multi-resolution searches were initialised from the mean model shape. The mean shape was aligned randomly within $\pm \pi/4$ of the best fit orientation for an annotation. The scale of the initialisation was set to lie within $\pm 25\%$ of the best fit scale, and the best fit position was offset by up to $\pm 20$ pixels in each direction. Five iterations of ASM search were carried out at each resolution level.

Table 6.4 shows a summary of model fit to nearest annotation distances for this set of experiments. The positions of the multi-resolution segmentations are substantially worse than those achieved using a single resolution model fitting strategy with the same initialisation position. The average distance between a model fit point and the nearest annotation boundary has increased from: single resolution $\mu_{all} = 16.0663$ to multi-resolution $\mu_{all} = 23.9146$. Robustness and accuracy have also been degraded.
Figure 6.11: Example wavelet and optimal discrimination texture image SASM segmentation results. Final model fits are shown in red, with annotations marked as yellow lines. (a), (c) and (e) show SASM search results on wavelet texture images, (b), (d) and (f) show SASM results on optimal discrimination texture images.
Chapter 6. Nerve Capillary Segmentation

Table 6.4: Model fit to annotation point-to-boundary distance summary, for SASM searches of wavelet texture images using single resolution, multi-resolution and genetic algorithm search.

<table>
<thead>
<tr>
<th>Method</th>
<th>$\mu_{\text{all}}$</th>
<th>$\sigma_{\text{all}}$</th>
<th>$\mu_{\text{acc}}$</th>
<th>$\mu_{\text{rob}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single resolution</td>
<td>16.0663</td>
<td>20.4655</td>
<td>1.1613</td>
<td>43.2894</td>
</tr>
<tr>
<td>Multi-resolution</td>
<td>23.9146</td>
<td>29.4852</td>
<td>1.3924</td>
<td>68.0707</td>
</tr>
<tr>
<td>Genetic algorithm</td>
<td>14.7676</td>
<td>16.4104</td>
<td>1.1407</td>
<td>38.8607</td>
</tr>
</tbody>
</table>

Figure 6.12: Histogram of multi-resolution SASM fit to annotation distances.

Figure 6.13: Histogram of genetic algorithm SASM fit to annotation distances.
Figure 6.14: Example multi-resolution and genetic algorithm SASM segmentation results. Final model fits are shown in red, with annotations marked as yellow lines. (a), (c) and (e) show multi-resolution SASM search results on wavelet texture images, (b), (d) and (f) show genetic algorithm SASM results on wavelet texture images.
Figure 6.12 shows a histogram of all model fit to annotation point–to–boundary distances. The tail of the distribution contains many mid to large values, indicating poor segmentation performance.

Table 6.4 also shows summary results from genetic algorithm segmentation experiments. The positions of the final boundaries are closer to an annotation than both the single and multi–resolution searches. Overall, the mean distance per point is \( \mu_{all} = 14.7676 \). The genetic algorithm searches also improve accuracy and robustness over both single and multi–resolution segmentations. These results have been gained in a fully automatic scheme, without using any specific prior knowledge for initialisation. Figure 6.13 shows a full histogram of segmentation boundary to annotation distances. Figure 6.14 shows some examples from the set of genetic algorithm segmentations.

### 6.3 Discussion

In this section, we put the results presented in section 6.2 in a wider context. We have two external standards by which segmentation results can be compared: the variance in expert annotations, and the data–driven snake segmentation results reported by Byrne [18].

The segmentation performance of ASMs and SASMs given ideal initialisations, resulted in overall mean point–to–boundary distances of \( \mu_{all} = 6.2292 \) and \( \mu_{all} = 5.5998 \) respectively. The histograms of the distances we noted to contain a small number of large outliers (figures 6.6 and 6.7). These outliers are caused by a single failure, which is shown in figure 6.8(e). The overall shape of the capillary in this image is extremely compressed along one dimension. In fact, there are no other examples in the training set that have this unusual shape. As the models have been built in a cross–validation scheme, neither the ASM or SASM represent this capillary shape as a valid class member. This causes segmentation failure for this image. Failures of
this type could be prevented if a larger set of training data was available.

If this single image segmentation is discounted, the results for SASM segmentations from an ideal initialisation would be: $\mu_{all} = 4.9125, \sigma_{all} = 5.5388, \mu_{acc} = 0.4676$ and $\mu_{rob} = 12.7220$. ASM results become: $\mu_{all} = 6.3073, \sigma_{all} = 8.9335, \mu_{acc} = 0.5051$ and $\mu_{rob} = 17.3517$. The difference between the ASM and SASM distributions without the single large failure remains significant at the 0.01 confidence level using either the Mann–Whitney or two sample Kolmogorov–Smirnov tests. The figures for the SASM segmentations are similar to the measured variability of expert annotations ($\mu_{all} = 4.0506, \sigma_{all} = 8.0642, \mu_{acc} = 0.3296$ and $\mu_{rob} = 12.1965$). However, SASM distances are significantly different to annotation variation, again using the Mann–Whitney and two sample Kolmogorov–Smirnov tests at the 0.01 significance level.

![Figure 6.15: Histogram of mean SASM segmentation to annotation distances for wavelet texture images.](image1)

![Figure 6.16: Histogram of mean SASM segmentation to annotation distances for optimal discrimination texture images.](image2)
Segmentation results achieved using wavelet and optimal discrimination features were compared in section 6.2. The difference between the two distributions of segmentation distances (figures 6.9 and 6.10) appeared to be a subtle one. If we present the same data in a slightly different way, differences in performance become clearer.

Each image segmentation has an average point-to-boundary error. This value is the mean of distances for all points on the segmentation boundary. This value can be calculated for each individual segmentation. Histograms of this average value for each set of segmentations are shown in figures 6.15 and 6.16. There is one large outlying failure in the set of segmentations using wavelet images. This one result corresponds to figure 6.11(e), which is the same unusual capillary mentioned above. In comparison, the optimal discrimination histogram shows six segmentations resulting in mid-range failures, the largest of which is again the same image as above (figure 6.11(f)).

One of the aims of this thesis is to compare model driven segmentation results with those obtained by Byrne [18] using carefully tuned Active Contour Models. The segmentations in Byrne’s study were generated using manually initialised paired snakes in a data-driven scheme. The paired snakes were initialised inside and outside the desired boundary, although the precise starting conditions were not recorded. Snakes initialisations were manually matched to the rough shape of each capillary. For our purposes, this can be thought of as aligning the snake with the gross capillary shape, and roughly estimating the scale and position of the capillary in the image.

The results presented in section 6.2.2 used initialisations that took into account some prior knowledge of capillary orientation, scale and position. Random offsets were used to reduce the influence of this prior knowledge. No specific prior knowledge of capillary shape was used. In the absence of exact manual starting positions used for the snake segmentations, we will assume that the amount of prior knowledge used to initialise each set of experiments was roughly equivalent. That is, we assume that we can directly compare the wavelet texture image SASM segmentation results with the
Chapter 6. Nerve Capillary Segmentation

data-driven snake segmentation results.

The average point-to-boundary error reported by Byrne was $\mu_{all} = 16.91$ pixels. The average error for SASM segmentations is $\mu_{all} = 16.03$ pixels. These measurements show that the model driven technique, which utilises prior knowledge to impose image interpretation constraints, slightly outperforms the data-driven approach. However, there are several more factors to take into account in this comparison.

Byrne’s results included a post processing step that refined the final segmentation boundary to fit more closely to local image evidence. This process has not been performed for the SASM segmentations. The evaluation method that Byrne employed (described in section 1.2) introduced a bias towards greater apparent accuracy. The SASM results have been evaluated using a set of annotations that have been generated independently. Byrne reports that five images resulted in clear segmentation failures. He did not include results from these images in his analysis. Expert clinicians have also rejected multiple images on more than one occasion during the annotation acquisition process. If we discard the single gross failure in the SASM results, segmentation results now become: $\mu_{all} = 14.2573, \sigma_{all} = 15.6304, \mu_{acc} = 1.1143$ and $\mu_{rob} = 37.0636$. Rejecting the worst five images would give a mean of $\mu_{all} = 12.2992$.

Multi-resolution image search has been found to give substantially worse results than single resolution segmentations. At low resolution levels, locally consistent, but misleading image evidence is indistinguishable from the main body of the capillary structure. Searches in the low resolution part of the image pyramid often become caught on these areas of misleading evidence. This causes initialisations at higher resolutions to be sufficiently inaccurate to result in poor final segmentations. Figure 6.14 shows three example multi-resolution segmentations. One image (figure 6.14(a)) contains no regions of misleading image evidence, and an accurate final segmentation has been achieved. In figures 6.14(c) and 6.14(e), significant regions of misleading evidence are present, and the resulting segmentations have become caught on them.

Genetic search experiments were also carried out to evaluate the performance of a fully
automatic segmentation scheme. The results achieved were: \( \mu_{all} = 14.7676, \sigma_{all} = 16.4104, \mu_{acc} = 1.1407 \) and \( \mu_{rob} = 38.8607 \), which represent an improvement over Byrne’s manually initialised snakes (snake results: \( \mu_{all} = 16.91 \)).

The improvement in automatic segmentation accuracy using genetic algorithms has been achieved at the cost of significantly increasing computational requirements. At least one iteration of SASM search was carried out for each individual at each generation of search. This compares to only five iterations carried out for each single resolution search.

Of the genetic algorithm searches, 34.38% have converged to a final segmentation with average distance less than 10 pixels. One example of a successful segmentation is shown in figure 6.14(b). However, the rest of the segmentations do not exhibit this degree of accuracy. In these cases, the algorithm has converged to its best segmentation position, but this does not lie on the desired boundary. Figure 6.14(d) shows an example where the global maximum fitness function is mostly positioned close to the desired boundary, but a substantial portion has deviated. In this case, the constraints imposed by the SASM model have not been sufficient to restrict the solution to the desired boundary in the face of good local image evidence. Occasionally, the fitness function is maximised in an entirely different position to the annotated BMEC boundary. This can be seen in figure 6.14(f). In this image, evidence around at the desired segmentation position is extremely weak, so it is not surprising that a global optimisation strategy will not give good results.

Overall, the results achieved by model driven automatic and manually initialised capillary segmentations have shown improvements over data driven segmentation techniques. Improvement gained through increasing model constraints, although statistically significant, is not large. Compared to expert annotation variation, segmentations are not robust enough to be used directly for reliable image measurement.


6.4 Summary

We have shown that SASMs including intermittent LEC boundary data result in more reliable segmentations than ASMs. The SASM is more a more specific model, and as such, imposes more constraints on the image segmentation problem. These extra constraints have been shown to increase the accuracy and robustness of capillary image segmentations. Incorporating prior knowledge in the form of a SASM statistical model also leads to improved segmentation results compared to a previous data-driven segmentation scheme.

The hypothesis that the complexity of optimal discrimination images would degrade model fitting performance has been supported by these segmentation experiments. Texture images generated using biorthogonal wavelet features have been shown to give better segmentation performance on this set of images.

Multi-resolution model fitting was found to perform poorly at low image resolutions. Initialisations at subsequent resolutions were therefore not accurate enough to allow convergence to a correct segmentation. For this reason, multi-resolution searches are not a suitable method to achieve automatic segmentation of nerve capillary images.

The performance of genetic algorithm model fitting is determined by the suitability of the fitness function to the segmentation task. The function we have chosen yields a good segmentation in the majority of cases. However, shape model constraints are not enough to restrict segmentations to the correct solution when multiple interpretations of the evidence exist.
Chapter 7

Conclusions and Discussion

The central aim of this thesis was to develop and utilise modelling techniques to constrain the automatic segmentation of capillary images. The highly variable mature of capillary appearance has meant that the constraints are weak. However, applying the weak constraints to this complex segmentation task has improved accuracy and robustness.

7.1 Modelling Capillary Shape and Appearance

7.1.1 Model Parameterisation

Capillary structures are extremely variable in shape and appearance. The lumen is not observable in every example. In chapter 3, we used the PDM formulation [36] as a ‘gold standard’ representation of capillary shape. This was done despite the lack of consistent anatomical features, using an ad hoc geometry–based landmarking scheme. Whole–boundary Fourier descriptor parameterisations [136] of capillary boundaries were found to result in far less compact and general models than PDMs. Furthermore, capillaries did not prove to be a suitable subject for appearance mod-
elling [30]. The large variability of the shape of capillaries caused serious distortions during the image warping process. These distortions create an unreliable model of capillary appearance, which does not perform well in model fitting.

### 7.1.2 Structured PDMs

Standard PDMs cannot represent the intermittently observed LEC boundary. In chapter 4, we developed a novel modelling scheme, capable of representing intermittent features. This scheme was called Structured PDMs [117], in which unobserved features are represented as missing data. The values of these missing data are estimated by a statistical imputation process. A standard PDM is then built using the completed data. The goal of the imputation process is to estimate values that retain the original relationships in the data, and introduce as little bias as possible into the future analysis. We have evaluated an iterated PCA imputation scheme that achieves this goal. This method has been shown to retain weak data relationships more fully than other statistical imputation methods.

A labelling of the structure of the missing data is produced and used, with shape parameters, to build a combined model of shape and structure. The combined model represents all shape and structure relationships in the original data. This modelling technique is general and applicable to any shape data where certain values are not observable in every example. The technique was extended to allow the appearance of intermittently present features to be modelled. We called this extension Structured Appearance Models.

The inclusion of extra LEC boundary data in a SPDM of capillary shape produces a more specific representation of the data (section 4.4.5). More specific models impose greater constraints on image interpretation. Evaluations have shown that using more highly constrained SASMs, rather than standard ASMs, improves segmentation performance in terms of overall average error, accuracy and robustness.
7.2 Model Fitting

Several modifications and extensions have been made to the ASM fitting algorithm. Nerve capillary images contain many areas of complex confusing evidence. This data often causes profile model best fit positions to deviate from the desired segmentation boundary. In chapter 3, we have shown that an error tolerant RANSAC model fitting scheme results in improvements in segmentation accuracy and robustness. The images also contain many areas of locally consistent, but confusing image evidence. Model fitting using the error–tolerant RANSAC method benefitted when the appearance of these regions was explicitly represented in a mixture model.

In chapter 6, model driven results were compared with results from a data–driven segmentation study. Constraints imposed by the use of prior shape knowledge were shown to improve overall segmentation accuracy given equivalent starting conditions. Multi–resolution image search, however, was found to be unsuitable for these images. Locally consistent, but misleading, areas of image evidence cause low resolution model fits to be inaccurate. Fully automatic segmentation can be achieved using global genetic algorithm model fitting. These searches resulted in the most accurate segmentations, but at increased computational cost dependent on the size of the genetic algorithms’ population and maximum number of generations. The performance of the global optimisation was limited by the suitability of the fitness function used to the specific segmentation task. In the majority of cases, fitness function maxima are found at a good segmentation position. However, occasionally, function maxima occur some way from the desired boundary.

In summary, we have shown that increasing constraints imposed on segmentation increase segmentation accuracy and robustness. The constraints imposed by shape models of capillaries are weak, but result in a measurable performance increase. Maximising constraints by utilising all available sources of information results in further improved segmentation performance. In complex segmentation tasks, the use of prior knowledge, even when shape and structure is highly variable, can improve image in-
terpretation.

7.3 Texture Analysis

The appearance of the nerve capillary images is highly variable. In chapter 5, texture analysis was used to simplify image evidence by maximising discrimination at the BMEC boundary. Biorthogonal wavelet texture features resulted in the lowest misclassification rate between image regions on the inside and outside of the BMEC boundary. An optimal discrimination scheme produced the best model fitting performance. However, these images were observed to contain a great deal of complex structure. This structure degrades segmentation performance when high quality initialisations are not available. Both measures improved on a previously developed texture analysis scheme [18] in terms of misclassification rate and model fitting performance.

7.4 Future Work and Open Questions

Several avenues for further study remain open at the end of this work. This section contains a very brief overview of potential directions for future work.

The models of the shape of capillaries can be further improved to impose greater constraints. More extensions and modifications to the various model fitting algorithms can also be made. The analysis of the texture content of the images can be extended to improve discrimination of the various capillary structures. Final model-based segmentations can be refined to fit to fine scale image evidence. Account must be taken of the reproducibility and consistency of the final measurements.
7.4.1 Modelling

If sub-sampled points are to be used as a basis for boundary modelling, there is considerable scope to optimise this representation. The *ad hoc* scheme used in this work could be improved upon using several published methods. Bookstein’s semi-landmarks [12] allow points to slide along a boundary to minimise a warping *bending energy*. This representation is optimal only with respect to this bending energy, not to boundary representation. Davies et al. [40] use an information theory formulation of model compactness to alter boundary parameterisations. This is a potentially useful method of determining optimal correspondence between boundaries.

In chapter 6, we outlined a problem with fitting incorrectly oriented models to image data. A potential solution lies in a modification to the Procrustes alignment algorithm to minimise point-to-boundary rather than point-to-point distance.

7.4.2 Texture Analysis

Texture analysis is an extremely wide area. There is still scope for further investigation of texture analysis methods. Specifically, further optimisation of the wavelet scheme presented in chapter 5 could be performed. It is possible to select the most significant branches of the wavelet decomposition tree [24]. Discarding unpromising features reduces memory requirements and allows a deeper exploration of the wavelet decomposition. It is also possible to use a continuous wavelet transform for texture analysis [122]. This might allow texture features to be focused closely on particular spatial frequency regions.

7.4.3 Boundary Refinement

In his snake based segmentation scheme, Byrne [18] added a final boundary refinement step. Snake boundaries were closely fitted to local image data using a dynamic
programming route optimisation through a cost graph. Models in this work have been built using parameterisations of smoothed structural boundaries. Therefore, this refinement process should be applied to the model driven segmentation results.

7.4.4 Reproducibility

To produce reliable statistics of capillary properties, clinicians require consistent, repeatable measurements. We note that both the RANSAC and genetic algorithm model fitting methods that we have employed are stochastic processes. However, the complexity of the images is so great that there is fundamental ambiguity in interpretation, even for expert clinicians. The effects of the stochastic nature of the model fitting process should be evaluated to determine if it is in line with the variation produced by experts.

7.5 Overall Performance

In this thesis, we have investigated the effects of modelling on capillary segmentation performance. We have shown that utilising prior knowledge improves performance. However, automatic segmentation results are still insufficiently robust. We define a ‘successful’ genetic algorithm segmentation to be a result with $\mu_{im} < 10$ pixels ($\mu_{im}$ is the average distance between each segmentation point on an image and the closest annotation boundary). Examples of ‘successful’ segmentations are shown in figure 7.1. Only 34.38% of segmentations fall into this successful group. Even on this measure, which appears to allow some segmentations of dubious quality, the robustness of the technique is not sufficient to be used directly for research purposes.

However, automatic segmentations may provide a starting point for an interactive scheme using model–driven methods. A live–wire type optimal path scheme [101] could be applied using model fit positions as seed points. These seed points could be
Figure 7.1: Examples of segmentations where $\mu_{im} < 10$ pixels.
manipulated by users leading to a fast, accurate and reproducible segmentation [6].

It is clear, when expert users are annotating images that the internal model is being adapted to the local image appearance. An optimal path scheme might be constructed that could apply such an adaptive model to the image structure.

7.6 Final Statement

The underlying goal of automatic capillary segmentation has not been solved by this work. However, a greater understanding of the benefits that can be achieved using statistical models of highly variable data has been gained.

Increasing the quantity of prior knowledge modelled increases the constraints that can be imposed on a segmentation task. Extra constraints lead to improved accuracy and robustness in the interpretation of complex images. Specifically, the inclusion of LEC boundary data using SPDMs measurably improves nerve capillary segmentations at the BMEC boundary.
Bibliography


