Minimizing the description length using steepest descent

Anders Ericsson and Kalle Åström
Mathematics, Center for Mathematical Sciences,
Institute of Technology, Lund University, Lund, Sweden
anderse@maths.lth.se

Abstract

Recently there has been much attention to MDL and its effectiveness in automatic shape modelling. One problem of this technique has been the slow convergence of the optimization step. In this paper the Jacobian of the objective function is derived. Being able to calculate the Jacobian, a variety of optimisation techniques can be considered. In this paper we apply steepest descent and show that it is more efficient than the previously proposed Nelder-Mead Simplex optimisation.

1 Introduction

Statistical models of shape [7, 15] have turned out to be a very effective tool in image segmentation and image interpretation. Such models are particularly effective in modelling objects with limited variability, such as medical organs.

The basic idea behind statistical models of shape is that from a given training set of known shapes be able to describe new formerly unseen shapes, which still are representative. The shape is traditionally described using landmarks on the shape boundary. A major drawback of this approach is that during training a dense correspondence between the boundaries of the shapes must be known. In practice this has been done by hand. A process that commonly is both time consuming and error prone.

There has been many suggestions on how to automate the process of building shape models, or more precise, finding a dense correspondence among a set of shapes [3, 5, 11, 12, 13, 17, 20, 23]. Attempts have been made to locate landmarks on curves using shape features, such as high curvature [5, 12, 20]. The located features have been used to establish point correspondences. Local geometric properties, such as geodesics, have been tested for surfaces [23]. Different ways of parameterising the training shape boundaries have been proposed [3, 13]. The above cited are not clearly optimal in any sense. Many have stated the correspondence problem as an optimisation problem [4, 6, 9, 10, 14, 18]. In [18] a measure is proposed and dynamic programming is applied to find the reparameterisation functions. A problem with this method is that it can only handle contours, for which the shape not changes too much, correctly. In [4] shapes are matched using shape contexts. In [2] the correspondence is located using proximity measures.

Minimum Description Length or MDL [16] is a paradigm that has been used in many different applications. In recent papers [8, 9] this paradigm is used to locate a dense

correspondence between the boundaries of shapes. It is a very successful algorithm. A problem with this method is, however, that the objective function is not stated explicitly and that it therfore has been hard to optimise. Nelder-Mead Simplex has been proposed. This optimisation technique is generally slow. In this paper we apply the theory presented in [19] and derive the gradient of the description length. We also propose an algorithm to minimize the description length (DL) using steepest descent.

This paper is organised as follows. In Section 2 the necessary background on shape models, MDL and calculating the gradient of the singular value decomposition is given. In Section 3, the gradient of the DL is derived and an algorithm to minimize the DL is proposed. In Section 4 we show that the convergence rate of the proposed algorithm is much faster than the effective algorithm proposed in [21] and that the models still are as accurate.

2 Preliminaries

2.1 Statistical Shape Models

When analysing a set of m similar (typically biological shapes) shapes, it is convenient and usually effective to describe them using Statistical Shape Models. Each shape is typically the boundary of some object and is in general represented by a number of landmarks. After the shapes $\mathbf{x}_i$ ($i = 0, \ldots, m-1$) have been aligned and normalized to the same size, a PCA-analysis is performed. A linear model of the form,

$$\mathbf{x}_i = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}_i, \tag{1}$$

can now describe the i-th shape in the training set. Here $\bar{\mathbf{x}}$ is the mean shape, the columns of $\mathbf{P}$ describe a set of orthogonal modes of shape variation and $\mathbf{b}_i$ is the vector of shape parameters for the i-th shape.

2.2 MDL

Let our shapes be represented by a number of parameterised curves $\mathbf{c}_i : [0,1] \mapsto \mathbf{R}^2$. We want to represent these curves by a linear shape model, as in (1). The problem of finding a dense correspondence among the shape boundaries is equivalent to reparameterising the shape boundary curves (to obtain $\mathbf{x}_i = \mathbf{c}_i \circ \gamma_i$), so that $\mathbf{x}_i(t)$ is the point that corresponds to $\mathbf{x}_j(t)$ for all $(i, j = 0, ..., m-1)$ and $t \in [0,1]$. Here $\gamma_i : [0,1] \mapsto [0,1]$ represents the reparameterisation of curve i. The same formulation can be used for, e.g. closed curves by changing the interval $[0,1]$ to the circle $\mathbf{S}^1$. MDL is a method to locate the parameterisation functions γ_i. The cost in MDL is derived from information theory and is, in simple words, the effort that is needed to send the model bit by bit. The MDL - principle searches iteratively for the set of functions γ_i that gives the cheapest model to transmit. The cost function makes a trade-off between a model that is general (can represent any instance of the object), specific (it can only represent valid instances of the object) and compact (it can represent the variation with as few parameters as possible). Davies and Cootes relate these ideas to the principle of Occam's razor: the simplest explanation generalises the best.

Since the idea of using MDL for landmark determination first was published [8], the cost function has been refined and tuned. Here we use the simple cost function stated in

[21]

$$DL = \sum_{\lambda i \geq c} (1 + \log \frac{\lambda_i}{c}) + \sum_{\lambda i < c} \frac{\lambda_i}{c}. \tag{2}$$

The scalar DL is the description length and is the cost to transmit the model according to information theory. The scalars λ_i are the eigenvalues of the linear model in equation (1) and c is a cut-off constant. Information can only be sent up to a certain degree of accuracy. The constant c expresses this accuracy. Typically we have set it to $c = 10^{-5}$, which corresponds to an acceptable error of 0.3 pixels for shapes with an original radius of 100 pixels.

There are two important properties of this cost-function. It is more intuitive than those formerly presented and the derivative is continuous.

2.3 Recapitulation of the SVD

In the rest of the paper, bold letters will be used for denoting vectors and matrices. The transpose of matrix $\mathbf{M}$ is denoted by $\mathbf{M^T}$ and m_{ij} refers to the (i, j) element of $\mathbf{M}$. The i-th non-zero element of a diagonal matrix $\mathbf{D}$ is referred to by d_i while $\mathbf{M}_i$ designates the i-th column of matrix $\mathbf{M}$. A basic theorem of linear algebra states that any real or complex $M \times N$ matrix $\mathbf{A}$ can be factored into the product of an $M \times M$ orthogonal matrix $\mathbf{U}$, an $M \times N$ diagonal matrix $\mathbf{S}$ with non-negative diagonal elements (known as the singular values), and an $N \times N$ orthogonal matrix $\mathbf{V}$.

In other words,

$$\mathbf{A} = \mathbf{USV^T} = \sum_{i=1}^{N} s_i \mathbf{U}_i \mathbf{V}_i. \tag{3}$$

The singular values are the square roots of the eigenvalues of the matrix $\mathbf{A}$

2.4 Computing the Jacobian of the singular values

In this Section the preliminaries of computing the Jacobian of the singular values are given. Here we recapitulate on the theory presented in [19]. For a more mathematical investigation in this field we recommend Alan Andrew's work, especially [1].

Employing the definitions of Section 2.3, we are interested in computing the derivatives of the singular values, $\frac{\partial d_k}{\partial a_{ij}}$ for every element a_{ij} of the $M \times N$ matrix $\mathbf{A}$. Taking the derivative of equation (3) with respect to a_{ij} gives the following equation

$$\frac{\partial \mathbf{A}}{\partial a_{ij}} = \frac{\partial \mathbf{U}}{\partial a_{ij}} \mathbf{SV^T} + \mathbf{U} \frac{\partial \mathbf{S}}{\partial a_{ij}} \mathbf{V^T} + \mathbf{US} \frac{\partial \mathbf{V^T}}{\partial a_{ij}}. \tag{4}$$

Clearly, $\forall (k, l) \neq (i, j), \frac{\partial a_{kl}}{\partial a_{ij}} = 0$, while $\frac{\partial a_{ij}}{\partial a_{ij}} = 1$. Since $\mathbf{U}$ is an orthogonal matrix, we have

$$\mathbf{UU^T} = \mathbf{I} \Rightarrow \frac{\partial \mathbf{U}^T}{\partial a_{ij}} \mathbf{U} + \mathbf{U^T} \frac{\partial \mathbf{U}}{\partial a_{ij}} = \omega_{\mathbf{U}}^{ij^T} + \omega_{\mathbf{U}}^{ij} = \mathbf{0}, \tag{5}$$

where $\omega_{\mathbf{U}}^{ij}$ is given by

$$\omega_{\mathbf{U}}^{ij} = \mathbf{U^T} \frac{\partial \mathbf{U}}{\partial a_{ij}}. \tag{6}$$

From Equation (5) it is clear that $\omega_{\mathbf{U}}^{ij}$ is an anti-symmetric matrix. Similarly, an anti-symmetric matrix $\omega_{\mathbf{V}}^{ij}$ can be defined for $\mathbf{V}$ as

$$\omega_{\mathbf{V}}^{ij} = \frac{\partial \mathbf{V^T}}{\partial a_{ij}} \mathbf{V}. \tag{7}$$

Notice that $\omega_{\mathbf{U}}^{ij}$ and $\omega_{\mathbf{V}}^{ij}$ are specific to each differentiation $\frac{\partial}{\partial a_{ij}}$. By multiplying Equation (4) by $\mathbf{U^T}$ and $\mathbf{V}$ from left and right respectively, and using Equations (6) and (7), the following is obtained

$$\mathbf{U}\frac{\partial \mathbf{A}}{\partial a_{ij}}\mathbf{V} = \omega_{\mathbf{U}}^{ij}\mathbf{S} + \frac{\partial S}{\partial a_{ij}} + \mathbf{S}\omega_{\mathbf{V}}^{ij}. \tag{8}$$

Since $\omega_{\mathbf{U}}^{ij}$ and $\omega_{\mathbf{V}}^{ij}$ are anti-symmetric matrices, all their diagonal elements are equal to zero. Recalling that $\mathbf{S}$ is a diagonal matrix, it is easy to see that the diagonal elements $\omega_{\mathbf{U}}^{ij}\mathbf{S}$ of and $\frac{\partial S}{\partial a_{ij}}\mathbf{S}\omega_{\mathbf{V}}^{ij}$ are also zero. Thus, Equation (8) yields the derivatives of the singular values as

$$\frac{\partial s_k}{\partial a_{ij}} = u_{ik}v_{jk}. \tag{9}$$

3 Method

In this Section the gradient of the description length is first derived and then an algorithm to minimize the description length is proposed.

3.1 Gradient of the Description length

In the proposed implementation each parameterisation function γ_i has n control nodes. Control node n on curve m is noted p_{mn}. The parameterisation function values in between the control nodes are evaluated by linear interpolation.

Differentiation of $\frac{\partial DL}{\partial p_{mn}}$ (2) gives

$$\frac{\partial DL}{\partial p_{mn}} = \sum_{\lambda_k \geq c} \frac{1}{\lambda_k} \frac{\partial \lambda_k}{\partial p_{mn}} + \sum_{\lambda_k < c} \frac{1}{c} \frac{\partial \lambda_k}{\partial p_{mn}}. \tag{10}$$

Here, the partial derivatives $\frac{\partial \lambda_k}{\partial p_{mn}}$ are needed. Let the m-th row vector of $\mathbf{X}$ be the configuration of landmarks for shape m when Procrustes analysis have been performed. By applying principal component analysis to $\mathbf{X}$, the shapes can be described with the linear model in equation (1). A singular value decomposition of $\mathbf{X}$ gives $\mathbf{X} = \mathbf{USV^T}$.

Here $\mathbf{V}$ corresponds to $\mathbf{P}$ in equation (1) and the diagonal $\mathbf{S^T S}$ gives the eigenvalues λ_k.

Now, if x_{mj} is the j-th landmark on shape m and $\frac{\partial x_{mj}}{\partial p_{mn}}$ is the derivative of the j-th landmark on shape m with respect to control node p_{mn} then

$$\frac{\partial \lambda_k}{\partial p_{mn}} = \frac{\partial s_k^2}{\partial p_{mn}} = 2s_k\frac{\partial s_k}{\partial p_{mn}} = 2s_k\sum_j \frac{\partial s_k}{\partial x_{mj}}\frac{\partial x_{mj}}{\partial p_{mn}}.$$

Putting in the results from Equation (9) gives

$$\frac{\partial \lambda_k}{\partial p_{mn}} = 2 s_k u_{mk} \mathbf{V}_m \frac{\partial x_{mj}}{\partial p_{mn}}.$$ (11)

In this implementation $\frac{\partial x_{mj}}{\partial p_{mn}}$ is calculated using differential approximation.

3.2 Algorithm

If the gradient of an objective function is known for a specific optimisation problem, it generally pays off to use more sophisticated optimisation techniques then Nelder-Mead Simplex or simulated annealing. Here steepest descent is proposed. An overview of the proposed algorithm is presented below.

Algorithm to minimise the description length

1. INITIALIZATION
   Initially the reparameterisation functions are set to arc-length parameterisation.

2. RESCALE AND ALIGN SHAPES
   The curves are aligned to the Procrustes condition

3. CALCULATE DL and dDL
   Calculate the gradient of the DL with respect to the parameterisation nodes p_{mn}

4. UPDATE PARAMETERISATIONS
   Search for a local minima in the gradient direction.

   (back to 2) until convergence

Algorithm 1: proposed algorithm to minimise description length

1) Initialization   Each shape is defined by a number of landmarks. Curves are defined between the original landmarks using linear interpolation. The curves are fixed during the whole optimisation. The only things that change are the reparameterisation functions. The parameterisation functions composed with the fixed curves define new curves. The landmarks that correspond to the original, fixed landmarks on the new curves are evaluated with the MDL-criteria.

2) Rescale and align Shapes   Each iteration starts by aligning and rescaling all curves according to the Procrustes alignment. When Procrustes is applied all landmarks are weighted equally. Therefore the Procrustes perform best if the landmarks are approximately equally distributed around the shapes. This is important to bear in mind.

3) Calculate DL and dDL   In this step the gradient according to Equation (11) is calculated for all parameterisation nodes. The gradient is evaluated after the Procrustes alignment has been performed. There seems to be no problems with this. It could be considered

to also optimise the alignment parameters, instead of aligning the shapes to the Procrustes condition.

4) Update parameterisations   A search for local minima is performed in the gradient direction. After an estimation of the local minima, all parameterisation nodes are updated at the same time. Once updated the algorithm starts over at 2) until convergence (roughly 50 iterations).

4   Experimental Validation

In this Section we validate our algorithm on five data sets, see Figure 1.

Hands   23 contours of a hand segmented out semi-automatically from a video stream. To simplify the segmentation the hand was filmed on a dark background.

Femurs   32 contours of femurs taken from X-rays in the supine projection.

Metacarpal   24 contours of metacarpals (a bone in the hand) deduced from standard projection radiographs of the hand in the posterior-anterior projection.

Silhouettes   The silhouette data set consists of 22 contours of silhouettes of faces. 22 persons were photographed using a digital camera. The silhouettes were then extracted using an edge detector.

The letter g   One data set of 17 curves of the letter g. The curves of the letter g are sampled using a device for handwriting recognition.

 On the five data sets the convergence speed of the proposed algorithm and the Nelder-Mead optimisation proposed in [21] is compared, see Figure 2. Thodberg's efficient implementation of MDL [21] has been used for the comparison. MATLAB source code and test data are available from `www.imm.dtu.dk/~hht`.

In all simulations, 9 control nodes have been used for the reparameterisations. Each curve is sampled with 64 landmarks to evaluate the description length at the given parameterisation. The initialisations for the Nelder-Mead and the steepest descent algorithm are identical.

To the left in Figure 2 the convergence rate (in seconds) of the description length using the two methods is plotted. It can be seen that the proposed optimisation scheme is much faster for all models.

There is one problem of the MDL approach. If all nodes are moved to approximately the same point on all curves, a very low description length is achieved. This can be prevented by using a master example. The master example is not reparameterised during optimisation. A node cost can also be applied as suggested in [21]. Local minima are another problem during optimisation. Due to these facts it is necessary to compare the quality of the models achieved using the two algorithms.
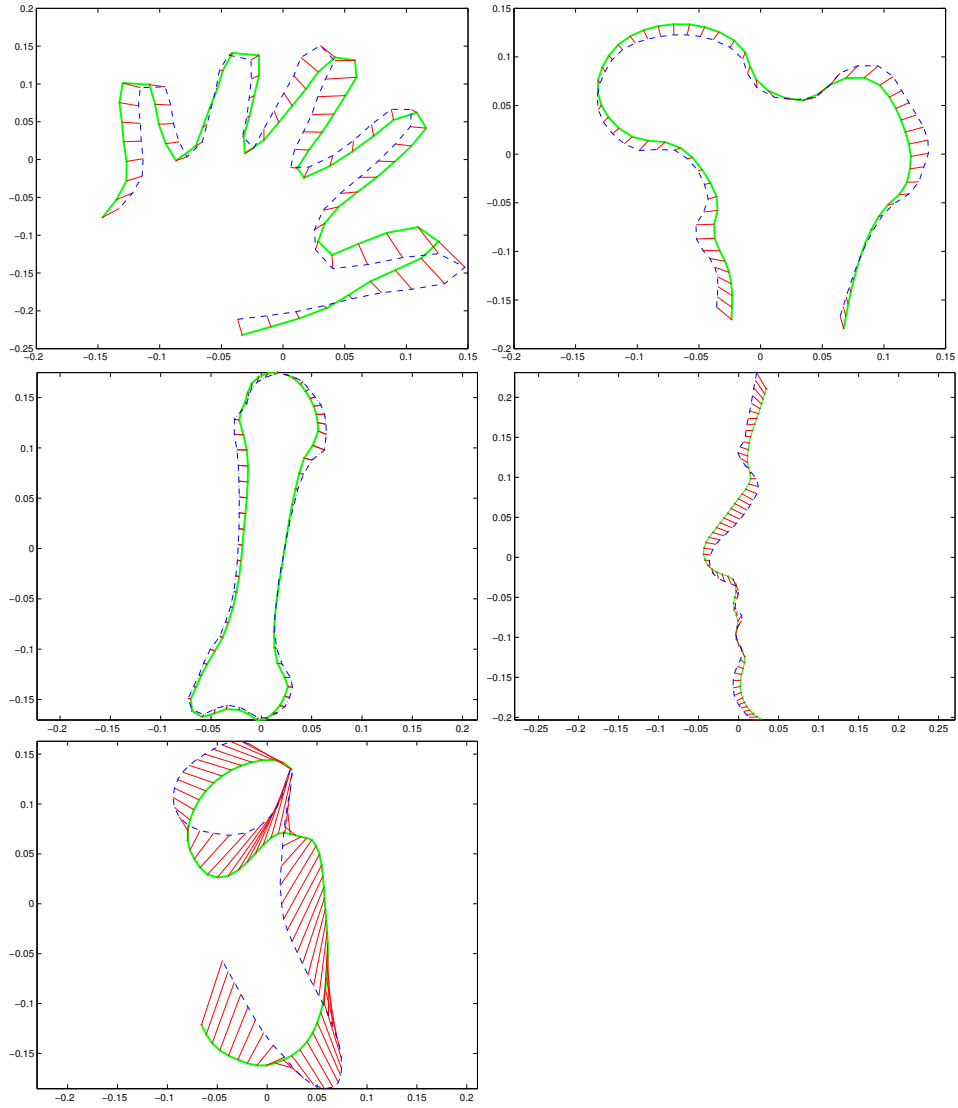
Figure 1: The mean (green solid line) and the first mode of variation (blue dashed line) of the optimised models (by the porposed algorithm) is plotted for the five datasets.
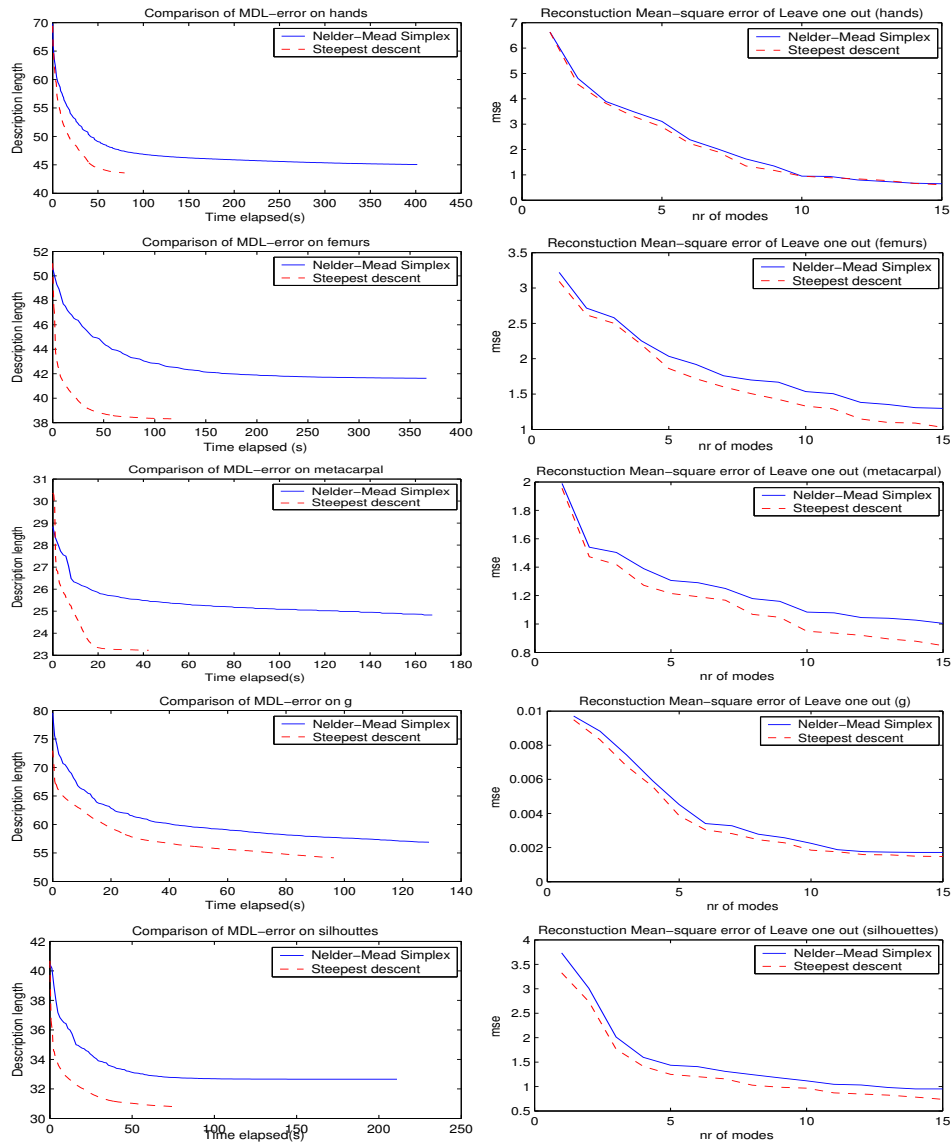
Figure 2: To the left we can see the convergence rate (in seconds) of the description length for the five models. To the right the mean squared approximation error of the five models is plotted against number of modes used. This measures the models ability to generalize.

The quality of the models is measured as the mean square error in leave-one-out reconstructions. The model is built with all but one example and then fitted to the unseen example. This is shown to the right in Figure 2. The plot shows the mean squared approximation error against the number of modes used. This measures the ability of the model to represent unseen shape instances of the object.

For all examples but the g:s we get models that generalize better using the steepest descent algorithm in one third to one forth of the time. For the g:s we run the proposed algorithm for 100 seconds before its ability to generalize is visibly better than the Nelder-Mead optimisation.

5   Summary and Conclusions

In this paper we present a more efficient way to minimize the description length. We derive the gradient of the description length and propose to use steepest descent to minimize the MDL-criteria. We have shown that the objective function is differentiable and can be written explicitly.

A result of applying steepest descent is that the objective function decreases in each iteration. The algorithm converges in just a few iterations and it is quite fast.

We have compared the proposed algorithm to the algorithm proposed in [22]. Better models are achieved for all cases. In four out of the five sets it takes one third to one forth of the time.

Acknowledgments

References

[1] Andrew A., Chu E., and Lancaster P. Derivatives of eigenvalues and eigenvectors of matrix functions. In *SIAM93 J. Matrix Anal. Appl.*, pages 903–926, 1993.

[2] Ericsson A. and Åström K. An affine invariant deformable shape representation for general curves. 2003 (to appear).

[3] A. Baumberg and Hogg D. Learning flexible models from image sequences. In *Proc. European Conf. on Computer Vision, ECCV'94*, pages 299–308, 1994.

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(24):509–522, 2002.

[5] A. Benayoun, Ayache N., and Cohen I. Adaptive meshes and nonrigid motion computation. In *Proc. International Conference on Pattern Recognition, ICPR'94*, pages 730–732, 1994.

[6] F.L. Bookstein. Landmark methods for forms without landmarks: Morphometrics of group differences in outline shape. *Medical Image Analysis*, 3:225–243, 1999.

[7] T.F Cootes and C.J. Taylor. *Statistical Models of Appearance for Computer Vision*. University of Manchester, 2001.

[8] Rhodri H. Davies, Tim F. Cootes, and Chris J. Taylor. A minimum description length approach to statistical shape modeling. In *Information Processing in Medical Imaging*, 2001.

[9] Rhodri H. Davies, Carole J. Twining, Tim F. Cootes, John C. Waterton, and Chris J. Taylor. A minimum description length approach to statistical shape modeling. *IEEE Trans. medical imaging*, 21(5):525–537, 2002.

[10] A. Hill and C. J. Taylor. Automatic landmark generation for point distribution models. In *Proc. British Machine Vision Conference*, pages 429–438, 1994.

[11] A. Hill and C. J. Taylor. A framework for automatic landmark indentification using a new method of nonrigid correspondence. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:241–251, 2000.

[12] C. Kambhamettu and D.B. Goldgof. Points correspondences recovery in non-rigid motion. In *Proc. Conf. Computer Vision and Pattern Recognition, CVPR'92*, pages 222–237, 1992.

[13] A. Kelemen, G. Szekely, and Gerig G. Elastic model-based segmentation of 3d neuroradiological data sets. *IEEE Trans. medical imaging*, 18(10):828–839, 1999.

[14] A.C.W. Kotcheff and C. J. Taylor. Automatic construction of eigenshape models by direct optimization. *Medical Image Analysis*, 2:303–314, 1998.

[15] V. Mardia, K. and I. L. Dryden. Shape distribution for landmark data. *Adv. Appl. Prob.*, pages 742–755, 1989.

[16] J. Rissanen. Modeling by shortest data description. *Automatica*, (14):465–471, 1978.

[17] D. Rueckert, F Frangi, and Schnabel J.A. Automatic construction of 3d-statistical deformation models using nonrigid registration. In *Medical Image Computing and Computer-Assisted Intervention MICCAI'2001*, pages 77–84, 2001.

[18] T. Sebastian, P. Klein, and B. Kimia. Constructing 2d curve atlases. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 70–77, 2000.

[19] Papadopoulo T. and Lourakis M. Estimating the jacobian of the singular value decomposition. In *Proc. European Conf. on Computer Vision, ECCV'00*, pages 555–559, 2000.

[20] H.D. Tagare. Shape-based nonrigid correspondence with application to heart motion analysis. *IEEE Trans. medical imaging*, 18:570–579, 1999.

[21] H. H. Thodberg. Minimum description length shape and appearance models. In *Image Processing Medical Imaging, IPMI 2003*, 2003.

[22] H. H. Thodberg. Minimum description length shape and appearance models. Technical Report IMM TECHNICAL REPORT 2003-01, Technical University of Denmark, 2003.

[23] Y. Wang, B.S. Peterson, and L.H Staib. Shape-based 3d surface correspondence using geodesics and local geometry. In *Proc. Conf. Computer Vision and Pattern Recognition, CVPR'00*, pages 644–651, 2000.