# MDL Spline Models: Gradient and Polynomial Reparameterisations

Jiří Hladůvka and Katja Bühler
VRVis Research Center for Virtual Reality and Visualization
Donau-City-Strasse 1, A-1220 Vienna, Austria
{Hladuvka|Buehler}@VRVis.at

**Abstract**

The major problem when building shape models by minimising the description length (DL) is the computationally demanding optimisation in a high-dimensional space.

To speed up the convergence when dealing with *discretised* contours, Ericsson and Åström [6] have shown how to *approximate* and how to exploit the gradient of the DL.

We derive the gradient of the DL for *differentiable* training sets. Additionally, we propose a class of polynomial reparameterisations that allows us to avoid numerical approximation of the functions and integrals involved in computation of the model and of its DL's gradient, making the whole process *exact* and more efficient.

## 1 Introduction

Deformable models find a variety of applications including segmentation and morphological analysis. An outstanding class of deformable shape models are statistical models which attempt to capture patterns of variability found in a class of objects [5].

A major requirement for building of such models is correspondence between the training boundaries. There are various attempts to automate search for the correspondence. Recently, algorithms based on minimising the description length of the outcoming model have established themselves as a gold standard in building of optimal shapes.

When establishing dense point correspondence on $n_s$ *discretised* boundaries, positions of $n_p$ points on each boundary are altered until an objective function based on the model's description length (DL) is minimised.

If the training shapes are defined as *parametric curves* [13, 2], we are interested in correspondence of their parameters rather than in resampling [5]. Each shape $S_i$ is reparameterised by a diffeomorphism $R_{\mathbf{w}_i}$ of the unit interval and the task is to manipulate the entire set $\{R_{\mathbf{w}_i}\}_{i=1}^{n_s}$ of reparameterisations until the description length is minimised.

Additionally to work of Davies [5] we assume the training boundaries to be differentiable. (If they are not, one can consider, e.g., yet another MDL-based algorithm of Cham and Cipolla [2] to achieve that.) To speed up the optimisation by gradient-based techniques, we derive the gradient of DL and propose a class of reparameterisations for its efficient computation.

BMVC 2005 doi:10.5244/C.19.90

In the next section we recall the continuous MDL models. From section 3 onwards we will assume that the training boundaries are differentiable. We introduce an expression to compute the Jacobian of eigenvalues of positive (semi)definite matrices and give a set of expressions to compute the *exact* gradient of the description length. In section 4 we propose a class of polynomial reparameterisations that allow an *analytic* and an *efficient* computation of integrals involved in the model and its gradient. In section 5 we summarize our concepts in two routines. We conclude in section 6 showing performance of our algorithm on two standard training sets.

## 2 Continuous MDL Models Revisited

In continuous models, the training boundaries $\{S_i\}_{i=1}^{n_s}$ are assumed to be $D$-dimensional parametric curves defined on the unit interval:

$$S_i : [0,1] \to \mathbb{R}^D, \qquad S_i(u) = (S_i^{x_1}(u), S_i^{x_2}(u), \ldots, S_i^{x_D}(u))^T, \qquad u \in [0,1] \tag{1}$$

Such curves form an infinite dimensional vector space with the inner product:

$$S_i \cdot S_j = \int_0^1 \sum_{d=1}^{D} S_i^{x_d} S_j^{x_d} \, \mathrm{d}u \tag{2}$$

Let $R_{\mathbf{w}_i}$ be a diffeomorphism of the unit interval determined by $n_w$ weights $\mathbf{w}_i = (w_{i1}, w_{i2}, \ldots, w_{in_w})^T$:

$$\begin{aligned} R_{\mathbf{w}_i(t)} &: [0,1] \to [0,1] \\ R'_{\mathbf{w}_i}(t) &> 0 \quad \forall t \in (0,1) \end{aligned} \tag{3}$$

Let $\mathbf{W}$ be an $n_w \times n_s$ matrix of weights for all shapes. Manipulating its elements will change the parameterisations of the training shapes. This will have impact on their correspondences, and consequently on their mean, variances and the outcoming statistical shape model.

In order to assess the quality of the outcoming model, its description length is addressed. The computation requires spectral decomposition of the training data covariance. As the covariance matrix of continuous shape data would be infinitely large, the spectral decomposition of a 'dual', real, symmetric, positively definite $n_s \times n_s$ matrix $\mathbf{C}$ is proposed instead [5]. The elements $c_{ij}$ of $\mathbf{C}$ are *inner* (rather than outer) products of the $i$-th and the $j$-th variances $V_i(t)$ and $V_j(t)$. Every element $c_{ij}$ is thus influenced by all $n_s$ reparameterised shapes $S_i(R_{\mathbf{w}_i})$ and by their continuous mean $\bar{S}$. Formally:

$$\bar{S}(t) = \frac{1}{n_s} \sum_{i=1}^{n_s} S_i(R_{\mathbf{w}_i}(t)) \tag{4}$$

$$V_i(t) = S_i(R_{\mathbf{w}_i}(t)) - \bar{S}(t) \tag{5}$$

$$\begin{aligned} c_{ij} = c_{ji} &= V_i \cdot V_j = \int_0^1 \sum_{d=1}^{D} V_i^{x_d}(t) V_j^{x_d}(t) \, \mathrm{d}t \\ &= \int_0^1 \sum_{d=1}^{D} \left( S_i^{x_d}(R_{\mathbf{w}_i}(t)) - \bar{S}^{x_d}(t) \right) \left( S_j^{x_d}(R_{\mathbf{w}_j}(t)) - \bar{S}^{x_d}(t) \right) \mathrm{d}t \end{aligned} \tag{6}$$

Since the training shapes $S_i$ are fixed, the elements $c_{ij}$ only depend on the weights. The matrix $\mathbf{C}$ can therefore be seen as a function of $\mathbf{W}$:

$$\mathbf{C} = C(\mathbf{W}), \qquad C : \mathbb{R}^{n_s n_w} \rightarrow \mathbb{R}^{n_s^2} \tag{7}$$

Once $\mathbf{C}$ is computed, its spectral decomposition $\mathbf{C} = \mathbf{ELE^T}$ is performed, yielding $n_s$ nonnegative eigenvalues $\lambda_1, \geq \ldots \geq \lambda_{n_s} = 0$ on the diagonal of $\mathbf{L}$, and $n_s$ orthonormal eigenvectors $\{\mathbf{e}_i\}_{i=1}^{n_s} = \mathbf{E}$. Vector $\lambda = (\lambda_1, \ldots, \lambda_{n_s})^T$ of the ordered eigenvalues can be seen as a function of matrix $\mathbf{C}$:

$$\lambda = \Lambda(\mathbf{C}), \qquad \Lambda : \mathbb{R}^{n_s^2} \rightarrow \mathbb{R}^{n_s} \tag{8}$$

To approximate the description length of the model using eigenvalues $\lambda$ [1], several objective functions have been proposed [7, 9, 5, 4, 14, 15]. In general, such an objective function is a real function of the $n_s$ eigenvalues:

$$O = O(\lambda), \qquad O : \mathbb{R}^{n_s} \rightarrow \mathbb{R} \tag{9}$$

As an example we recall the recently introduced differentiable objective function of Thodberg [14] due to a threshold variance $\lambda_c > 0$ that reflects the granularity of the training set:

$$O(\lambda_1, \ldots, \lambda_{n_s}) = \sum_{\lambda_i \geq \lambda_c} (1 + \log(\lambda_i / \lambda_c)) + \sum_{\lambda_i < \lambda_c} \lambda_i / \lambda_c \tag{10}$$

Composition of the functions $O$, $\Lambda$, and $C$ yields the description length $DL$ of the model, that assesses its quality, hence the underlying correspondence of boundaries:

$$DL = DL(\mathbf{W}) = O \circ \Lambda \circ C(\mathbf{W}), \qquad DL : \mathbb{R}^{n_s n_w} \rightarrow \mathbb{R} \tag{11}$$

To find the optimal model (i.e. the optimal correspondence of the training set), the description length $DL$ has to be minimised with respect to the weights $\mathbf{W}$. To find the global minimum of $DL$ in this $n_w n_s$–dimensional space is a computationally demanding task and a significant effort has been spent for its efficient completion [5, 14, 6].

# 3 Gradient of DL for Differentiable Shapes

Whenever it is possible to compute the gradient of function to be minimised, it is worth considering a gradient–based search technique. Assuming that the training shapes $S_i(u)$ are differentiable in $u$ and that the functions $R_{\mathbf{w}_s}$ are differentiable in $w_{sk}$, all of the functions $O$, $\Lambda$, $C$, and therefore $DL$ are differentiable and the gradient $\nabla DL$ exists.

As the $DL$ is a composition of differentiable functions, its gradient $\nabla DL$ can be computed by applying the generalized chain rule in multidimensions. Differentiation of Eq. (11) yields a product of three Jacobian matrices of dimensions $(1 \times n_s)$, $(n_s \times n_s^2)$, and $(n_s^2 \times n_s n_w)$:

$$\nabla DL = J_O \cdot J_\Lambda \cdot J_C \tag{12}$$

Let us have a closer look at the Jacobians:

[1] In this work we do not address the full and a more complex DL based on the projection of the training set to eigenvectors [5].

**The $1 \times n_s$ Jacobian of $O$ :** The Jacobian of the objective function (10) is computed easily:

$$\frac{\partial O}{\partial \lambda_s} = \begin{cases} 1/\lambda_s & \text{if } \lambda_s \geq \lambda_c \\ 1/c & \text{if } \lambda_s < \lambda_c \end{cases} \qquad s = 1, \ldots, n_s \tag{13}$$

**The $n_s \times n_s^2$ Jacobian of $\Lambda$ :** As $\mathbf{C}$ is positive semidefinite, all its eigenvalues are non-negative and its spectral decomposition $\mathbf{C} = \mathbf{ELE}^{\mathbf{T}}$ is a special case of the singular value decomposition (SVD). The rate of the change of the SVD's diagonal matrix $\mathbf{L}$ with respect to elements of $\mathbf{C}$ was derived by Papadopoulo and Lourakis [11]. We point out that their theorem can be applied directly to eigenvalues of positive (semi)definite matrices:

$$\frac{\partial \lambda_s}{\partial c_{ij}} = e_{is} e_{js} \qquad i, j, s = 1, \ldots, n_s \tag{14}$$

It should be noted that we are applying the Jacobian of SVD in a different way and for a different matrix than Ericsson and Åström [6] who did it for rectangular variance matrices of discretised training sets.

**The $n_s^2 \times n_s n_w$ Jacobian of $C$ :** What remains to find is how the elements of the matrix $\mathbf{C}$ change with the $k$-th weight of the $s$-th shape reparameterisation, i.e., how the terms $\partial c_{ij}/\partial w_{sk}$ look like for $i, j, s = 1 \ldots n_s$, $k = 1 \ldots n_w$. Differentiation of (6) yields:

$$\begin{aligned} \frac{\partial c_{ij}}{\partial w_{sk}} &= \frac{\partial}{\partial w_{sk}} \int_0^1 \sum_{d=1}^{D} \left( S_i^{x_d}(R_{\mathbf{w}_i}) - \bar{S}^{x_d} \right) \left( S_j^{x_d}(R_{\mathbf{w}_j}) - \bar{S}^{x_d} \right) \mathrm{d}t \\ &= \int_0^1 \frac{\partial}{\partial w_{sk}} \sum_{d=1}^{D} \left( S_i^{x_d}(R_{\mathbf{w}_i}) - \bar{S}^{x_d} \right) \left( S_j^{x_d}(R_{\mathbf{w}_j}) - \bar{S}^{x_d} \right) \mathrm{d}t \\ &= \frac{1}{n_s} \int_0^1 \sum_{d=1}^{D} \frac{dS_s^{x_d}}{dR_{\mathbf{w}_s}} \frac{\partial R_{\mathbf{w}_s}}{\partial w_{sk}} \left( (\delta_{is} n_s - 1)\, V_j^{x_d} + (\delta_{js} n_s - 1)\, V_i^{x_d} \right) \mathrm{d}t \end{aligned} \tag{15}$$

where $\delta_{is}$ and $\delta_{js}$ denote the Kroneker symbol.

# 4 Polynomial Reparameterisations

The way how the integrals (6) and (15) may be computed strongly depends on the form of reparameterisation functions $R_{\mathbf{w}_s}$.

Davies *et al.* proposed parameterisations by taking an integral of a weighted sum of Gauss [3] or Cauchy [5] kernels of different positions, widths and heights. These functions involve the function $\mathrm{Erf}(t)$ or $\mathrm{ArcTan}(t)$ and, when back-substituted into (6) and (15), force the integrals to be approximated numerically, because the primitive functions can not be found.

Assuming that the differentiable training shapes are polynomial functions (e.g. splines [2, 8]), we propose polynomial reparameterisations. If both $S_i$ and $R_{\mathbf{w}_s}$ are polynomials, all the differentiation, composition and multiplication terms in Eqs. (6), (15) are polynomials. The integrands as well as their primitive functions are therefore polynomials, too. The integrations decimate thus to evaluations of these polynomials in integration limits that can be computed exactly and, employing Horner's rule [12], efficiently.

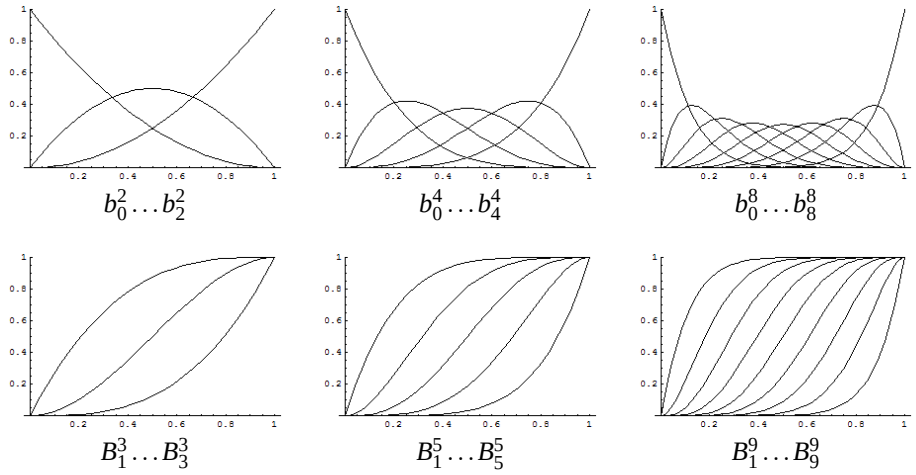


Figure 1: Bernstein polynomials (top row) of degrees 2, 4, and 8 and their normalized polynomial integrals of degree 3, 5, and 9 (bottom row).

The purpose of this section is to construct *polynomial* reparameterisations $R_{\mathbf{w}_s}$ and their derivatives, given the $n_w$ weights $w_{sp}$.

Inspired by work of Davies and collaborators [3, 5] we intend to take an integral of a positive function on (0,1). Instead integrating a combination of distribution functions (such as Gauss or Cauchy kernels), we propose to take an integral of a linear combination of Bernstein basis polynomials restricted to the unit interval (Fig. 1).

The $k$-th Bernstein polynomial of degree $n$ is defined as:

$$b_k^n(t) = \binom{n}{k} t^k (1-t)^{n-k} \qquad n \geq 0 \quad k = 0, \dots, n \tag{16}$$

On interval $[0,1]$ these basis polynomials are non-negative, attain their maximum at $t = k/n$, and sum up to 1.

Let us consider a linear combination of Bernstein polynomials with nonnegative weights $w_{sk}$:

$$r_{\mathbf{w}_s}(t) = \frac{1}{\sum_{p=1}^{n_w} w_{sp}} \sum_{p=1}^{n_w} w_{sp}\, b_{p-1}^{n_w-1}(t) \qquad w_{sp} \geq 0 \quad \mathbf{w}_s \neq \mathbf{0} \tag{17}$$

Being restricted to open interval $(0,1)$, $r_{\mathbf{w}_s}(t)$ is a positive polynomial of degree $n_w - 1$. Its integral is therefore a monotonically increasing polynomial of degree $n_w$ and (discarding the integration constant) can easily be shown to map the unit interval to $[0, 1/n_w]$. Its multiplication by $n_w$ yields therefore the desired diffeomorphism of $[0,1]$.

We define the reparameterisation of the unit interval due to $n_w$ nonnegative weights $\mathbf{w}_s$ as follows:

$$\begin{aligned} R_{\mathbf{w}_s}(t) \stackrel{def.}{=} n_w \int r_{\mathbf{w}_s}(t)\mathrm{d}t &= n_w \int \frac{1}{\sum_{p=1}^{n_w} w_{sp}} \sum_{p=1}^{n_w} w_{sp}\, b_{p-1}^{n_w-1}(t)\ \ \mathrm{d}t \\ &= \frac{1}{\sum_{p=1}^{n_w} w_{sp}} \sum_{p=1}^{n_w} w_{sp} \int n_w\, b_{p-1}^{n_w-1}(t)\ \ \mathrm{d}t \\ &= \frac{1}{\sum_{p=1}^{n_w} w_{sp}} \sum_{p=1}^{n_w} w_{sp}\, B_p^{n_w}(t) \end{aligned} \tag{18}$$

where the $n_w$ polynomials $B_p^{n_w}$ of degree $n_w$ (Fig. 1) can be precomputed using the following closed form:

$$B_p^{n_w}(t) = \sum_{d=1}^{n_w} (-1)^{d+p} \binom{d-1}{d-p} \binom{n_w}{d} t^d \qquad p = 1 \ldots n_w \tag{19}$$

As $B_1'(0) = B_{n_w}'(1)$, the concept can easily be extended to a differentiable parameterisation of closed boundaries keeping $w_1 = w_{n_w}$ and 'gluing' the ends together. An identity reparameterisation is obtained setting $w_{s1} = w_{s2} = \ldots = w_{sn_w} > 0$ (Fig. 2). Looking back at Figure 1 it can also be seen that our parameterisations are naturally suited for a hierarchical refinement of the reparameterisation. At level $L > 1$, the maxima of polynomials $b_k^{2^L}$ are distributed evenly on [0,1] and are placed right in between the maxima of $b_k^{2^{L-1}}$.

Differentiation of $R_{\mathbf{w}_s}(t)$ with respect to the $k-th$ parameter $w_{sk}$ yields again an $n_w$-th degree polynomial in $t$:

$$\frac{\partial R_{\mathbf{w}_s}(t)}{\partial w_{sk}} = \frac{1}{(\sum_{p=1}^{n_w} w_{sp})^2} \sum_{p=1}^{n_w} w_{sp}(B_k^{n_w}(t) - B_p^{n_w}(t)) \qquad k = 1, \ldots, n_w \tag{20}$$

Substitution of the terms $R_{\mathbf{w}_s}$ and $\partial R_{\mathbf{w}_s}/\partial w_{sk}$ back into Eqs. (6) and (15) leads to the desired integration of polynomials.

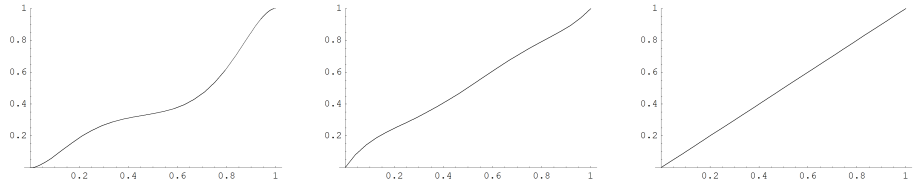


Figure 2: Example reparameterisations by 9 weights. The more the weights are similar the more the reparameterisations approach the identity. The weight vectors from left to right: $\mathbf{w} = (0, 3000, 0, 0, 0, 0, 0, 6000, 0)^T$, $\mathbf{w} = (0.899,\ 0.006,\ 0.139,\ 0.580,\ 0.166,\ 0.855,\ 0.125,\ 0.117,\ 0.640)^T$, and $\mathbf{w} = (\alpha, \alpha, \alpha, \alpha, \alpha, \alpha, \alpha, \alpha, \alpha)^T, \alpha > 0$.

# 5 Implementation

When running steepest descent or conjugate-gradient optimisation, sequences of minimisation along a line in directions opposite to gradient are performed. While necessary when line minimisation reaches minimum, gradients are not necessarily needed when performing the line minimisation itself. Even though that the gradient calls occurred by only 4–6 % of the *DL* calls it should be taken into consideration that the gradient calls are more expensive – its computations requires eigenvectors, computation of moderate matrices, and their multiplication. Therefore we suggest to have separate routines and avoid the gradient calls for line minimisation. The two separate routines are outlined in Algorithms 1 and 2.

**Algorithm 1** Computation of the DL in $\mathbf{W_0}$

| | |
|---|---|
| Construct $R_{w_s}$ | ▷ Eq. (18) |
| $\mathbf{C}_0 \leftarrow C(\mathbf{W}_0)$ | ▷ Eq. (6) |
| $\lambda_0 \leftarrow \Lambda(\mathbf{C}_0)$ | ▷ get *only* eigenvalues of $\mathbf{C}_0$ |
| $DL \leftarrow O(\lambda_0)$ | ▷ Eq. (10) |
| **return** $DL$ | |

**Algorithm 2** Computation of the $\nabla DL$ in $\mathbf{W_0}$

| | |
|---|---|
| Construct $R_{w_s}$ | ▷ Eq. (18) |
| $\mathbf{C}_0 \leftarrow C(\mathbf{W}_0)$ | ▷ Eq. (6) |
| $\mathbf{C}_0 \rightarrow \mathbf{ELE^T}$ | ▷ get *both* eigenvalues *and* eigenvectors of $\mathbf{C}_0$ |
| $\mathbf{C}_0{}' \leftarrow J_C(\mathbf{W}_0)$ | ▷ Eq. (15) |
| $\lambda_0{}' \leftarrow J_\Lambda(\mathbf{C}_0)$ | ▷ Eq. (14) |
| $\mathbf{O}_0{}' \leftarrow J_O(\lambda_0)$ | ▷ Eq. (13) |
| $\nabla DL \leftarrow \mathbf{O}_0{}' \lambda_0{}' \mathbf{C}_0{}'$ | ▷ Eq. (12) |
| **return** $\nabla DL$ | |

As the search space might potentially contain many local minima, we propose, inspired by work of Ku *et al.* [10], to roughly bracket the global minimum by genetic algorithm prior to an optimisation by a local search technique.

# 6 Results

The quality of models based on objective function (10) in terms of one-leave-out tests, generalisation ability, specificity, and compactness has been proven elsewhere [14, 6] and we do not include such an evaluation in the results. Instead we concentrate on the improvement in performance achieved by our concepts.

We implemented the computation in C++, used GALib [1] for genetic algorithm and compared the conjugate-gradients versus (the gradientless) Nelder-Mead simplex routines of Numerical Recipes [12] in number of *DL* calls. Such a comparision is legitimate since the number of $\nabla DL$ calls contributed by only 4–6 % of the *DL* calls on average and did not affect the overall performance significantly.

We demonstrate the functionality and the performance of our concepts on two well-known training sets:

**Synthetic Bump** Similarly to Davies we have used (the hard to optimise) synthetic bump as a proof of concept. The training set consists of 10 bumps of different positions. Bumps were modeled by 6 polynomial segments in tension [8]. To model the corners by differentiable functions, high node tension factors have been set.

The only mode of variation (Fig. 3) has been correctly found already by quadratic reparameterisations ($n_w = 2$). Performance comparison of conjugate-gradient versus Nelder-Mead simplex search in terms of number of *DL* calls is summarized in Figure 4.

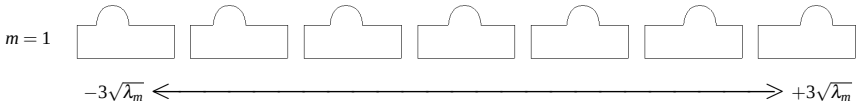


Figure 3: Synthetic bump. The only mode of variation.

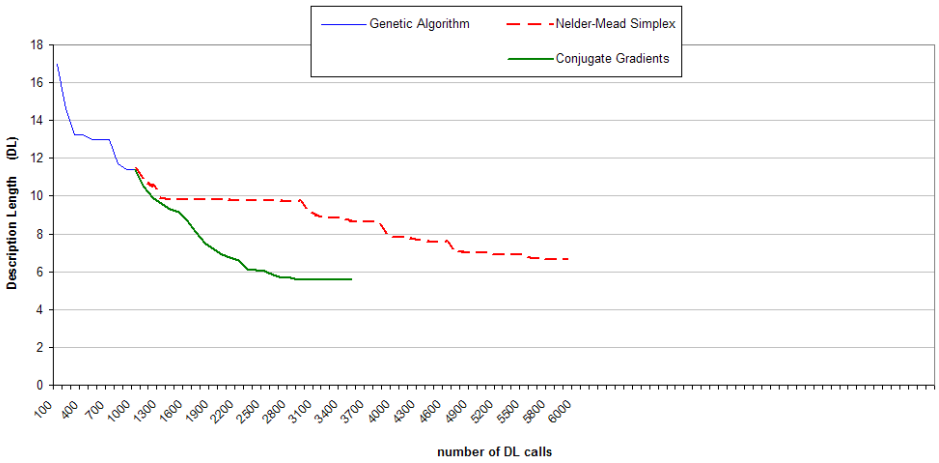


Figure 4: Performance chart for the training set of bumps.

**Hands** This training set consists of 16 hands modeled using splines in tensions [8]. To approach the polygonal shape that appears frequently in the literature, the node tensions were set 1000-times higher than the segment tensions. All of the splines consist of 56 segments. Figure 5 shows the first four modes of variation found by 4-th degree polynomial reparameterisations ($n_w = 4$). Performance of conjugate-gradient versus Nelder-Mead simplex search in terms of number of *DL* computations is summarized in Figure 6.

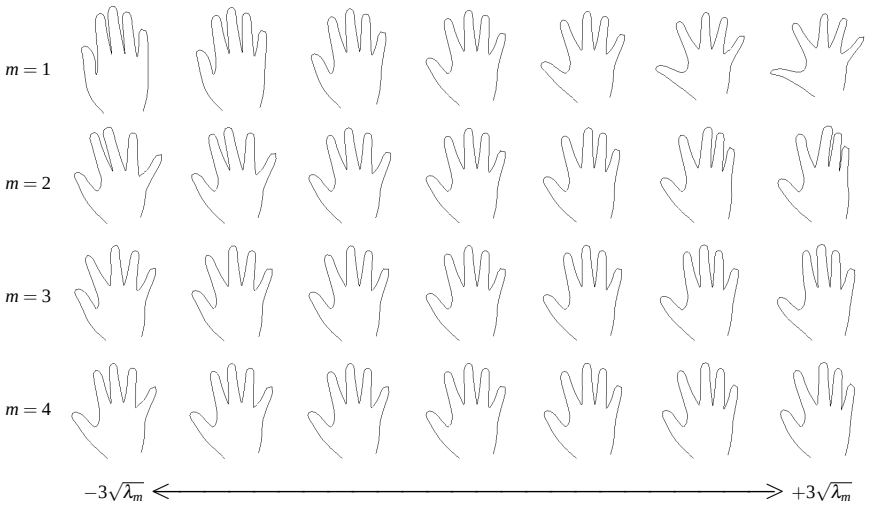


Figure 5: Hand. The first 4 modes of variation.

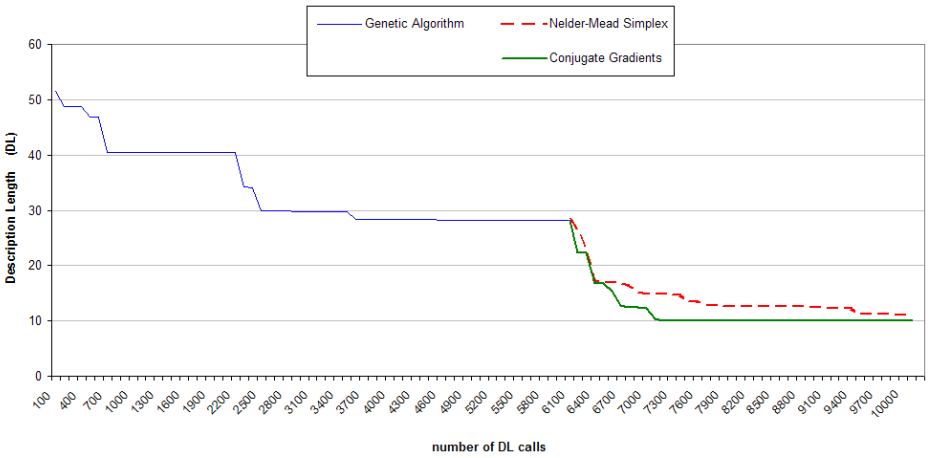


Figure 6: Performance chart for the training set of hands.

# 7 Conclusion and Future Work

We have derived the gradient of the description length for training shapes with differentiable boundaries. The most critical part in this derivation was the Jacobi matrix of eigenvalues. Inspired by previous work [11, 6] we have shown how this can be carried out trivially for positive definite matrices that are involved in computation.

Furthermore, we have proposed a class of reparameterisation functions that allow an exact and an efficient computation of both the model and the gradient of its description length.

Newly introduced concepts have been tested on 2 standard training sets and the performance tests indicate a potential for practical applications.

In the near future we would like to extend the polynomial reparameterisations to boundaries of 3D objects.

# 8 Acknowledgements

The hand data set was kindly provided by Hans Henrik Thodberg.

# References

[1] GAlib: A C++ library of genetic algorithm components. http://lancet.mit.edu/ga/.

[2] T.J. Cham and R. Cipolla. Automated B-spline curve representation incorporating MDL and error-minimizing control point insertion strategies. *IEEE PAMI*, 21(1):49–53, 1999.

[3] Rh. H. Davies, T.F. Cootes, C.J. Twining, and C.J. Taylor. An information theoretic approach to statistical shape modelling. In *Proceedings of British Machine Vision Conference (BMVC)*, 2001.

[4] Rhodri H. Davies, Tim F. Cootes, and C. J. Taylor. A minimum description length approach to statistical shape modelling. In *Proceedings of Information Processing in Medical Imaging (IPMI)*, 2001.

[5] Rhodri Huw Davies. *Learning Shape: Optimal Models for Analysing Shape Variability*. PhD thesis, University of Manchester, 2002.

[6] A. Ericsson and K. Åström. Minimizing the description length using steepest descent. In *Proceedings of British Machine Vision Conference (BMVC)*, 2003.

[7] A. Hill and C. J. Taylor. Automatic landmark generation for point distribution models. In *Proceedings of British Machine Vision Conference (BMVC)*, pages 429–438, 1994.

[8] Josef Hoschek and Dieter Lasser. *Grundlagen der geometrischen Datenverarbeitung*, chapter 3.6.2 Polynomiale Splines in Tension, pages 101–105. B. G. Teubner Stuttgart, 1992.

[9] A. C. W. Kotcheff and C. J. Taylor. Automatic construction of eigenshape models by direct optimisation. *Medical Image Analysis*, 2(4):303–314, 1998.

[10] Kim W. C. Ku, M. W. Mak, and W. C. Siu. Approaches to combining local and evolutionary search for training neural networks: A review and some new results. In S. Tsutsui and A. Ghosh, editors, *Advances in Evolutionary Computing*, pages 615–642. Springer, 2003.

[11] T. Papadopoulo and M. Lourakis. Estimating the Jacobian of the singular value decomposition. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 555–559, 2000.

[12] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 2 edition, 1992.

[13] D. G. Schweikert. An interpolation curve using a spline in tension. *Journal of mathematics and physics*, 45:312–317, 1966.

[14] H. H. Thodberg. Minimum description length shape and appearance models. In C. Taylor and J. A. Noble, editors, *Proceedings of Information Processing in Medical Imaging (IPMI 2003)*, Lecture Notes In Computer Science (LNCS) 2732. Springer, 2003.

[15] H. H. Thodberg and H. Olafsdottir. Adding curvature to minimum description length shape models. In *Proceedings of British Machine Vision Conference (BMVC)*, 2003.