Adding Curvature to Minimum Description Length Shape Models

Hans Henrik Thodberg and Hildur Olafsdottir
Informatics & Mathematical Modelling
Technical University of Denmark
2800 Lyngby, Denmark
hht@imm.dtu.dk
http://www.imm.dtu.dk/~hht

Abstract

The Minimum Description Length (MDL) approach to shape modelling seeks a compact description of a set of shapes in terms of the coordinates of marks on the shapes. It has been shown that the mark positions resulting from this optimisation to a large extent solve the so-called point correspondence problem: How to select points on shapes defined as curves so that the points correspond across a data set. However, this MDL approach does not capture important shape characteristics related to the curvature of the curves, and occasionally it places marks in obvious conflict with the human notion of point correspondence. This paper shows how the MDL approach can be fine-tuned by adding a term to the cost function expressing the mismatch of curvature features across the data set. The method is illustrated on silhouettes of adult heads. The MDL method is able to solve the point correspondence problem and a classification of the heads into male and female improves dramatically when using the MDL-generated marks.

Keywords: point correspondence problem, minimum description length, shape modelling, curvature, face recognition, silhouettes.

1 Introduction

1.1 The Point Correspondence Problem

This paper addresses the point correspondence problem in statistical shape modelling, which arises when 2D or 3D shapes are generated in terms of a curve or surface with no landmarks. The problem is how to define points all over the curves or surfaces so that the points correspond across the set.

First we notice that the solution depends on what we mean by "correspondence". A number of possible qualities of a point assignment can be enumerated:

1. A suitable class of experts should agree with the point placements.

2. The points should lead to good performance of some subsequent processing, e.g. a classification task based on the shape points, or a pattern recognition task e.g. an active shape or active appearance model based on the points.

3. Cross validation: How well does a statistical shape model based on the defined points describe the points on new unseen shapes? This can be measured by leave-one-out cross validation.

4. The points should to be faithful to the shapes, i.e. the points should not "cheat" by avoiding some parts of the shape or by placing points too thinly, so that the details are not reflected in the point coordinates.

5. The point selection algorithm should behave consistently in the limit of infinitely densely placed points.

6. The algorithm should be practical, i.e. fast and with few parameters to tweak.

7. The algorithm should generalise from 2D to 3D.

8. Compactness of the shape model of the set as measured by the MDL cost function or a similar measure.

In 2001-2003 the Minimum Description Length (MDL) approach received a lot of attention [4],[5],[6],[8], because it scores well on many of the virtues above, and this paper elaborates on the MDL framework starting by an analysis of these virtues:

Item 1 has not been addressed in the MDL literature and is the subject of the present paper.

Item 2 was addressed by [7] who compared the use of MDL with manual marks in the subsequent AAM analysis. The use for classification is addressed in this paper in Section 3.2.

Item 3 was examined in [4],[5],[6]. It is not easy because the new case should be fitted to the model, i.e. its marks should be optimised before the evaluation can be made.

Item 4 was addressed in [6] which chose a master example with fixed points. It was further analysed in [8] which proposed adding a controlling term to the cost imposing a constraint on the average position of the marks.

Item 5 was addressed in [6] and [8]

Item 6 was addressed in [8] which published a fast Matlab code with a single essential parameter, the desired accuracy of the modelling

Item 7 was addressed in [4], [5],[6].

Item 8 is a surrogate quality and sort of self-fulfilling and was used by [4],[5],[6] to show how much better the MDL result is compared to the manual annotation.

To summarise, items 1-3 are the fundamental goals, items 4-7 are technical properties, while 8 - as explained below - is not an actual goal.

1.2 Ockhams's Razor

The philosophy of Ockhams's Razor is that the simplest or most compact description is truer. So the basic assumption is that if we select the points on the curves so that we obtain the simplest model, then the virtues 1-3 will be automatically fulfilled to a high degree. High simplicity is interpreted as small description length, and the description length is

a functional of the eigenvalue spectrum. Of course one cannot prove Ockhams's Razor; ultimately inference is based on assumptions which cannot be proved. So item 8 should not be regarded as a virtue - it is not the goal; it is rather the meta-principle utilised to obtain the real goals 1-3.

This paper focuses on how well the MDL approach addresses item 1 and 2, and how the approach can be enhanced in this respect.

1.3   The Rehabilitation of Curvature

Often experts identify points of extreme curvature as landmarks. But in the original formulation of the MDL shape models, curvature was somewhat discredited as being less "fundamental":

> *Shape features (e.g. regions of high curvature) have been used to establish point correspondences, with boundary length interpolation between these points. Although this approach corresponds with human intuition, it is still not clear that it is in any sense optimal.* [4]

This paper reconciles the use of MDL with the use of shape features. This is done by introducing an explicit curvature representation in the MDL formalism.

The standard MDL method seeks compact description of the *positions* of the shape points. But there is more information in a curve than its positions, and the curvatures are proposed as another salient piece of information. By requiring the model to describe both *positions* and *curvatures* we get a different optimum with point correspondences that matches both positions and curvatures.

This state of affairs is not unusual in image analysis: segmentation can be intensity-based as in AAM [1], or edge-based as in ASM [2], and the results will in general differ. The two features can be combined in a common optimisation by adding edge features to AAM [3]. Intensity-based methods are in general more robust, owing to their larger basin of attraction and they are less susceptible to noise, but edge-based methods often agree more accurately with the expert opinions.

One of the problems with curvature-defined landmarks is that in biological shapes these points are not always born out clearly or uniquely by the individual shape, and in this case the landmarks must be inferred from a more global context. Thus the problem calls for a way to gracefully combine the position-defined correspondences with curvature signatures, which are not the same across the set, but varies according to a statistical model.

2   The Model

The description of the models falls in five sections:
1. The dynamic variables - the nodes.
2. The MDL cost of the mark positions.
3. The node cost stabilising the configurations.
4. The curvature cost.
5. The optimisation strategy.

2.1 The Dynamical Variables: The Nodes

The method was described in detail in [8] and is briefly reviewed here.

The algorithm applies to a set of shapes defined as curves in 2D space. Shape sets are classified into three kinds: Closed curves, open curves with fixed end-points and open curves with free end-points. The arc length along the curve is normalised to run from 0 to 1.

We are seeking a set of $2^L + 1$ marks on each curve, where L is an integer, to represent the shape. For closed shapes, the start- and end-points (number 0 and 2^L) are identical. The mark locations are specified in a hierarchical manner on L levels. For closed curves with 65 marks, we specify on the first level the coordinates of mark 0 and 32 by their absolute arc length position. On the second level, mark 16 and 48 are specified by parameters between 0 and 1. For example mark 16 can be anywhere on the curve between mark 0 and 32, corresponding to the extremes 0 and 1. On the third level the marks 8, 24, 40 and 56 are specified in between already fixed marks. This is continued until level 6 so that all marks are specified.

For open *fixed*-end curves, level 1 places only mark 32, while for open *free*-end curves there are three marks on level 1, namely 0, 32 and 64.

The end-marks are defined by two positive parameters describing the distance of the end-marks from the curve ends.

The initial shape can be defined by marks placed evenly in arc length by setting all parameters to $a = 0.5$ (except for the end-marks). Alternatively a priori knowledge of a good starting guess can be used. On closed curves mark 0 should be approximately aligned initially.

To save computation, the optimisation is usually only done on a subset of marks, these active marks are called *nodes*, depending on the level of detail. The optimisation adjusts the node parameters to optimise the correspondence of all the marks over the set of examples. The parameters of the passive marks are frozen at 0.5 corresponding to even distribution in arc length.

2.2 The MDL Cost of the Mark Positions

Statistical shape analysis is performed on the mark positions in the usual way. The number of marks is $N = 2^L$ for closed curves and $N = 2^L + 1$ for open curves (free as well as fixed). First the shapes are centred and aligned to the mean shape normalised to one. The covariance matrix of the aligned shapes is formed and principal component analysis is performed yielding the eigenvalue spectrum. The objective function is defined from the eigenvalue spectrum and a parameter λ_{cut}.

$$
\begin{aligned}
&\text{Description Length} = \sum L_m \\
&L_m = 1 + \log(\lambda_m/\lambda_{\text{cut}}) && \text{for } \lambda_m \geq \lambda_{\text{cut}} \\
&L_m = \lambda_m/\lambda_{\text{cut}} && \text{for } \lambda_m < \lambda_{\text{cut}}
\end{aligned}
\tag{1}
$$

This cost is continuous at λ_{cut} and independent of N in the large-N limit.

2.3 The Node Cost of the Mark Positions

Some mechanism must be introduced to prevent that marks pile up in some regions and dilute in others. [6] suggested fixing the marks of a master example, and this works

approximately OK, but still means the master can become an outlier in the resulting statistical distribution.

A more satisfactory method was introduced in [8]. One introduces a target a_i^{target} for the *average* parameter a_i^{average} for each node i by means of a quadratic cost:

$$\text{NodeCost} = \sum (a_i^{\text{average}} - a_i^{\text{target}})^2 / T^2 \tag{2}$$

where T is a chosen tolerance.

2.4 The Curvature Variation Cost

Curvature is computed along the shape as follows

$$t_i = r_{i+1} - r_{i-1}$$
$$c_i = \pi N (r_{i+1} - r_{i-1} - 2r_i) \cdot \hat{f}_i / t_i^2 \tag{3}$$

where r_i is the 2-vector coordinates of point i, t_i is the tangent, and $\hat{f}_i$ is the normal. This curvature expression is independent of the pose of the shape and it is 1 for a circle. The curvatures c_i are then smeared with a Gaussian filter, which in this study has sigma 1.5 (this should scale with N). The smeared curvature value at mark i of for example r is denoted k_{ir}. For open curves the curvature cannot be computed at the ends, and close to the ends it also becomes noisier due to the smearing. Therefore curvature near the ends is not included, and for the silhouette data set below using 64+1 marks, 5 curvatures at each end are skipped, i.e. 55 curvatures are used.

The curvatures could now be weighted with a factor and appended to the aligned position coordinates and included in the PCA. However, a simpler method is used here. The following extra term to the cost function is constructed to measures the compactness of the curvature description of the set:

$$\text{CurvatureCost} = C \frac{1}{N} \frac{1}{s} \sum_{i,r} (k_{ir} - k_i^{mean})^2$$
$$k_i^{mean} = \frac{1}{s} \sum_r k_{ir} \tag{4}$$

Here s is the number of shapes and C is a weighting factor for this term. The curvature cost is independent of the resolution, as the other terms in the cost function.

This simple curvature model states that all shapes have the same curvature signature, and the cost measures the deviation from the mean.

2.5 Optimisation Strategy

The iterative optimisation can then begin. The nodes, e.g. 8, are ordered according to ascending level. Each node is associated with a step length, initially set to 0.01. These 8 step lengths are automatically decreased by the algorithm. Now the parameters a(node) for each node and each example are probed, one at a time according to the following pseudo-code, which runs over a number of passes, until the results has stabilised.

```
Loop over passes
   Loop over nodes
      Loop over 5 steps
         Loop over examples
            Loop over + and - step
               Probe a(node) = a(node) +- step of example
               Recompute marks of example
               Do Procrustes of set
               Do PCA of set
               Compute new MDL
               If new MDL is lower, accept and break loop
               Undo a(node) change
            End of +- step loop
         End of example loop
         If < 5% of a(node)'s changed, divide step(node) by 2
         If >20% of a(node)'s changed, multiply step(node) by 2
      End of step loop
   End of node loop
End of passes loop
```

Intensity-based methods are often more robust than edge-based methods, and likewise we expect position- based shape matching to be more robust than curvature-based. Curvature is a more futile feature; a given curvature feature can sometimes be weaker or even disappear, and curvature is more vulnerable to noise. Therefore the curvature term is used only in the later stages of optimisation to fine-tune the result of the two other cost terms, i.e. the following strategy is used:

1) Optimise with MDL position cost, i.e. without curvature cost.
2) Reset step lengths to 0.01.
3) Optimise with total cost including curvature cost.
4) Optionally repeat from 2 using a larger weight C on the curvature cost.

3   Results

3.1   The Silhouette Case

22 silhouettes of heads shown in Figure 2 are used as a test case. No master example is used - instead the node cost is used (with T=0.05) so that the curves are represented by marks, which on average are evenly distributed in arc length (see [8] for details). The start and end locations of the shapes are also optimised, and here the node cost requires that an average of 12% is skipped in the top, and 7% in the bottom. 18 nodes parameters are optimised, chosen more finely near the lips and the eyes.

$\lambda_{\text{cut}} = 0.03^2$ is used, and the first 4 passes are performed without curvature term leading to the result in Figure 1.

Then 6 passes are run with curvature cost with $C = 150$, and this leads to the remarkably accurate assignment of points seen in Figure 2. The curvatures before and after the

inclusion of curvature cost are shown in Figure 3. At convergence, MDL cost= 35.85, curvature cost= 18.96 and node cost= 1.79. The entire optimisation takes 5 minutes on a 1.2 GHz PC under Matlab. If the curvature cost is turned off again, and the optimisation continued, the solution stays close to the solution in Figure 2 and reaches MDL cost = 31.60, 3.13 lower than in Figure 1 - which thus is a local minimum. Hence the curvature cost also works as a catalyst for minimising the MDL cost.
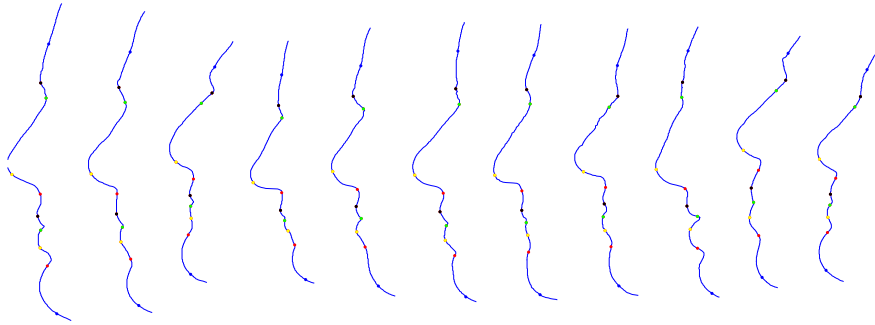

Figure 1: MDL without curvature cost: Notice the problems at the top of the nose, and at the joining of the lips. The points are the extremes of the curvature of the mean shape after convergence expressed as interpolations of the 65 shape points. In addition, the start and end of the shapes found by the algorithm are indicated by points.
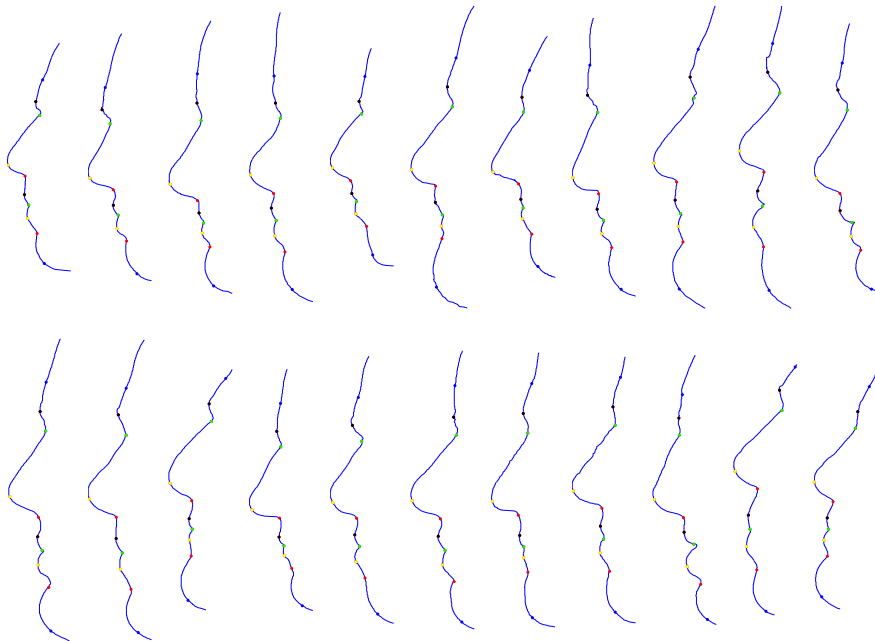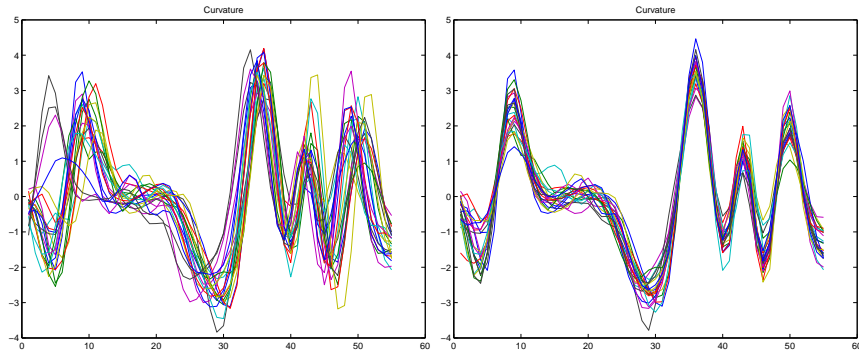

Figure 2: Result of MDL with curvature cost

Figure 3: Smeared curvature before and after optimisation with the curvature cost. The before-plot shows three examples with bad upper limit of the nose.

3.2 Classification of silhouettes

A quantitative comparison of marks placed by MDL and manually placed marks is performed according to the following analysis. An additional 17 silhouettes are collected, yielding in total 20 male and 19 female adults. A manual annotation of the silhouettes is performed and an automatic MDL placement is generated.

A statistical model is made to predict the gender of the profile using logistic regression on a subset of the PCA scores as inputs, determined by backwards elimination using a classical statistical test for significance. The p-values of the manual and MDL models are determined. In addition a leave-one-out cross-validation of the regression models is performed. The variable selection is also part of the leave-one-out process. The number of correct classification is presented in Table 1. Finally 8 human observers are asked to classify the silhouettes. The number of correct evaluations and standard deviation among the 8 evaluations are recorded as shown in Table 1.

	p-value	Percentage correct
Manual marks	0.03	65
MDL marks	0.00003	85
Direct human scoring	-	65 ± 5

Table 1: Gender determination using various models.

From Table 1, it is seen that MDL gives significantly better models of the gender than manual marks, and it outperforms by far the direct human observers. It was found important to include the chin of the faces to obtain this good performance (approximately as much as in Figure 2) - as men have larger chins. For this reason, care was taken to ensure that the two sets of marks on average cover the same amount of the front and the chin. The direct human scoring uses the full curve, i.e., more than covered by the marks, but despite of this, it performs worse. Figure 4 shows a few examples of classification using MDL annotation.
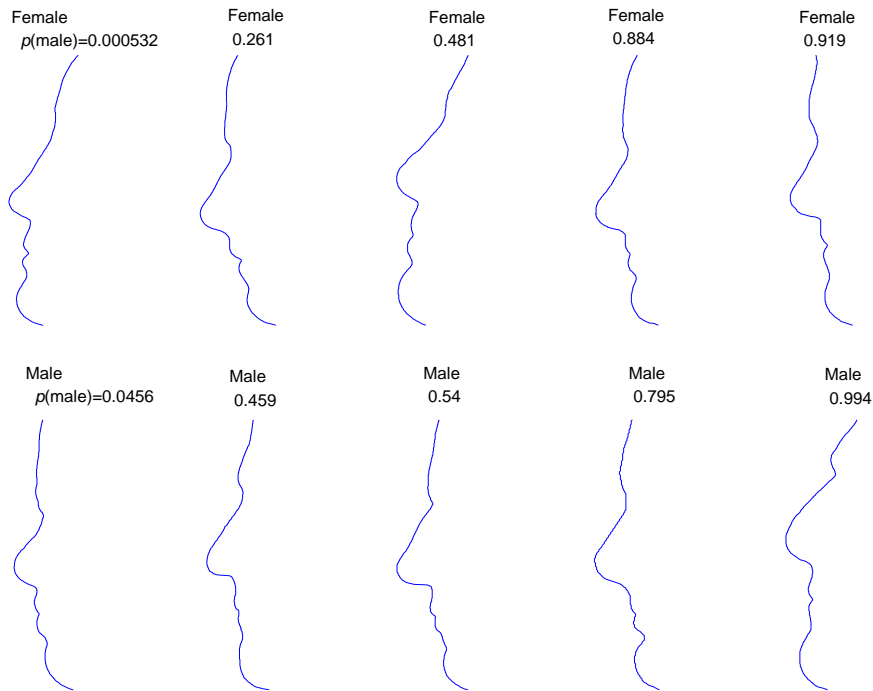
Figure 4: Classification of silhouettes using MDL annotation. True gender and p(male) from logistic regression model.

4   Discussion and Conclusion

The curvature at mark i is defined solely from r_{i-1}, r_i and r_{i+1} so it may seem that the curvature is redundant information. However, it amplifies second-order derivative content of the curve, which plays a negligible role in the PCA of the positions. So the curvature digs out more information from the curves. Of course this should only be used when the data is known to have sufficiently low noise to allow for this.

The method unifies the intuitive idea of matching points with high curvature with the MDL approach yielding automation, robustness and accuracy. Computer vision rarely allows for a unique and perfect solution; instead the method seeks an optimum, which is a trade- off between competing and frequently contradictory requirements, and controlled by the weighting factor C of the curvature term.

The vision of the approach is that simplicity implies truth. Now we can see that this truth depends on the input: Adding curvature explicitly changes the simplest solution. So in order to find the truth, your representation of the data must be chosen carefully. If the curve is noisy, curvature is not a true feature of the data. So the moral of the story is that care is needed when defining the representation.

As for the faces the noise is not in the curvature but rather in long-range x-y-positions: Noise appears when the viewing of the head is not strictly lateral, but somewhat from above or slightly towards frontal. Finally the distance to the head gives perspective distortions. These three degrees of freedom affects the location more than the curvatures.

The feature vector can either be position and curvature described by a common PCA, or it can be two separate cost functions of position and curvature respectively. In the first case the model can exploit correlations between positions and curvature and form a compact description. In the latter case curvature and position are required to be compact by themselves. This paper explored the division of cost into two terms which is easy to control and computationally fast, and it leads to a solution of the point correspondence problem for the silhouette case in excellent agreement with human experts and based on a principled approach. Classification of shapes was greatly improved using the MDL method.

Jesper Skjerning is acknowledged for providing the first 22 silhouettes.

The Matlab software source code is available from www.imm.dtu.dk/~hht.

References

[1] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. In *European Conference on Computer Vision*, volume 2. H. Burkhardt and B Neumann Ed, Springer, 1998.

[2] T.F. Cootes and C.F. Taylor. Active shape models - smart snakes. In *British Machine Vision Conference*. Springer-Verlag, 1992.

[3] T.F Cootes and C.J. Taylor. On representing edge structure for model matching. In *IEEE CVPR*, volume 1, pages 1114–1119, 2001.

[4] R.H. Davies, T.F. Cootes, and C.J. Taylor. A minimum description length approach to statistical shape modelling. *IPMI*, 14, 2001.

[5] R.H Davies, C.J. Twining, T.F. Cootes, J.C. Waterton, and C.J. Taylor. 3d statistical shape models using direct optimisation of description length. In *7th European Conference on Computer Vision, ECCV*. Heyden, A. et al editors, Springer, 2002.

[6] R.H. Davies, C.J Twining, T.F Cootes, J.C Waterton, and C.J. Taylor. A minimum description length approach to statistical shape modelling. *IEEE Trans Med. Imaging*, 21:525–537, 2002.

[7] M.B. Stegmann and R.H. Davies. Corpus callosum analysis using mdl-based sequential models of shape and appearance. *submitted to Medical Image Computing and Computer-Assisted Intervention - MICCAI*, 2003.

[8] H.H. Thodberg. Minimum description length shape and appearance models. *accepted for 16th IPMI*, 2002.