Strings and necklaces: on learning and browsing medical image segmentations
Ghebreab, S.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Chapter 3

Strings: Variational Deformable Models of Multivariate Ordered Features*

We propose a new image segmentation technique called strings. A string is a variational deformable model that is learned from a collection of example objects rather than built from a priori analytical or geometrical knowledge. An object boundary is represented by a one-dimensional curve in multivariate functional space rather than by a point as is the case in active shape models. In the learning phase, multiple shape and image feature functions along continuous object boundaries in a learning set are aligned, then subjected to functional principal components analysis and functional principal regression to model the feature space. A Mahalanobis distance model takes the natural variations in the learning set into account for evaluation of boundaries. In the segmentation phase, an object boundary in a new image is searched for with help of a deformable feature function, a string. The string is weighted by the regression model and evaluated by the Mahalanobis model. A curve is deformed in the image to produce feature functions with minimal distance. Strings have been compared with active shape models on 145 vertebra images, showing that strings produce better results when initialized close to the target boundary, and comparable results otherwise.

*Conditionally accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence
3.1 Introduction

In advanced image segmentation problems object boundaries frequently have inhomogeneous characteristics. The shape of a target boundary may be blunt at some parts and strongly convoluted at other parts. Or, the image gradient along a target boundary may be clearly visible and pointing outwards at some places, while it is hardly defined due to neighboring objects at other places. We note that in many advanced segmentation problems boundaries are fractured, occluded, convoluted or inhomogeneous otherwise, requiring the definition of multiple features for accurate description of the boundary. For this reason, it is imperative to construct inhomogeneous boundary models for application in image segmentation. We learn such boundary models by exploring the information contained in large collections of example images rather than constructing them from analytical or geometrical knowledge rules.

Learning in the context of boundary-based image segmentation has received considerable attention in literature, e.g. in [12], [43], [58], [71], [114], [108], in particular within the active shape model framework [22]. In active shape model methods, a boundary is learned by statistical analysis of feature values from a set of example boundaries. During image segmentation, the learned boundary model is used as a reference for deformation of an active shape and for evaluation of a boundary recorded by that shape. The generic approach of active shape models is very appealing. However, they often fail to fully exploit the multivariate continuous characteristics of a boundary. The question that is raised here is how multiple boundary features such as edge gradient and contour curvature, are exploited for learning a continuous variational image segmentation model.

We propose a unified approach to learning structurally different boundary features for multi-feature image segmentation. The problem of learning is transposed into one of analyzing the closed functional curves in feature space that best describe average feature values and the most important variations therein. Image segmentation is conceived of as an iterative procedure of recording multiple continuous feature values, weighting these feature values to amplify the statistically most descriptive features and evaluating the weighted features values with respect to the values seen in the learning set. The proposed segmentation approach combines theory from functional data analysis [95] with theory from chemometrics [69] to arrive at string segmentation models.

The chapter is organized as follows. Section 2 discusses previous work on active shape models. Section 3 introduces the string method. First the learning phase will be described in detail, then image segmentation with help of deformable strings is addressed. Implementation issues, experiments and results are described in section 4. Discussion and conclusion follow in section 5.

3.2 Related Work

A number of active shape models have been presented in literature. The method introduced by Cootes et al. in [22] makes use of established statistical techniques to
3.2. Related Work

construct a shape model from examples. In the reference, a shape is represented as an $N$-vector of vertices

$$\mathbf{x} = [(x_1, y_1), ..., (x_N, y_N)]^T.$$  \hfill (3.1)

Assuming two-dimensional vertices, each $N$ vertex is a single point in a $2N$-dimensional vector space. The set of $M$ shape examples subsequently forms a learning set of size $M$

$$\mathcal{L} = \{\mathbf{x}_1, ..., \mathbf{x}_M\}.$$  \hfill (3.2)

To remove variation from the learning set attributed to stretching, shearing and rotation of the shapes, the example shapes are aligned by Procrustes analysis [50], aiming at minimizing the sum of distances of each example shape to the average

$$\epsilon_{\mathcal{L}} = \sum_{m=1}^{M} ||\mathbf{x}_m - \bar{\mathbf{x}}||^2$$  \hfill (3.3)

where $\bar{\mathbf{x}}$ is an initial estimate of the average shape, with $||\bar{\mathbf{x}}|| = 1$. In an iterative procedure, the $M$ shape examples are aligned one-by-one, with each iteration refining the estimation of the average shape.

Assuming the cluster of aligned shapes forms an ellipsoid, in [22] principal components analysis is performed to reduce the dimensionality of the data using the covariance matrix

$$\mathbf{C}_\mathcal{L} = \frac{1}{M-1} \sum_{m=1}^{M} (\mathbf{x}_m - \bar{\mathbf{x}})(\mathbf{x}_m - \bar{\mathbf{x}})^T$$  \hfill (3.4)

where $\bar{\mathbf{x}}$ now denotes the average shape of the aligned example shapes. The eigenvalues $\lambda_n$ of $\mathbf{C}_\mathcal{L}$, with $\lambda_n \geq \lambda_{n+1}$ for $n = 1, ..., 2N$, tell the amount of variance captured by each principal component. The largest fraction of the total variance is given by the first $Q \leq 2N$ eigenvalues

$$\lambda = \sum_{q=1}^{Q} \lambda_q.$$  \hfill (3.5)

The eigenvalues $\lambda_q$ describe the most significant modes of variation and the corresponding eigenvectors describe the dimensions in which they occur. A shape instance $\mathbf{x}$ is explained as the mean shape plus some linear combination of these eigenvectors

$$\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{P} \mathbf{b}^T$$  \hfill (3.6)

where the $Q \times 2N$ matrix $\mathbf{P}$ contains the first $Q$ eigenvectors and $\mathbf{b}$ is a $Q$-vector of weighting coefficients. New plausible shapes are generated by varying $\mathbf{b}$ within suitable limits, derived by statistically examining the distribution of the weighting coefficients required to generate the learning data. On the basis of these feasible
shapes high gradient boundaries are searched for in the image.

A number of shortcomings of the active shape model in [22] have been recognized and dealt with in literature.

One problem arises when the learning set $\mathcal{L}$ is contaminated with outliers, influencing the statistics adversely. For this reason, Dutta et al. [35] remove outliers in an iterative approach using an inter-shape distance matrix that defines the mean alignment error between a polygonal approximation of a shape instance and an original shape from the learning set. The reduced learning set forms the basis for the construction of a less distorted statistical shape model. Application of the corrected shape model for image segmentation consequently leads to better performance. We adopt this idea and we remove outliers in one step using another distance measure.

Another problem is that minimization of equation 3.3 only takes into account pose and scale differences between instances and does not account for non-linear shape differences. This is solved by Dutta et al. [35] using a flexible point matching technique that performs global similarity registration of two arbitrary sets of points and non-linear registration based on local similarity of two curves. Instead of minimizing equation 3.3, the trade-off between a compensated mean alignment error and the number of correspondences is minimized. This way the effect that unconstrained linear registration of two sets of points tends to shrink [42] one set with respect to the other is avoided. In addition, in [35] no correspondence between points is required when performing non-linear registration. Other work on the shape alignment problem includes the work by Kotcheff et al. [68], who optimize model compactness rather than variance. We also aim at linear and non-linear registration while avoiding problems with the cardinality of point sets.

A restriction of point distribution models is that deformation of the shape model by adjusting weighting vector $b$ only allows limited deformation reflecting the variations in the learning set. For this reason, Wang et al. [125] use prior models based on principal component analysis of additional covariance matrices. By replacing equation 3.4 with a weighted and mixed covariance matrix that deals with independence and smoothness they are capable of building a wide range of shape models even when there are few examples shapes or if the learning set exhibits small variation. An earlier attempt to add artificial variation to the statistical model is found in [21]. We take a different road aiming at building models also capable of explaining objects dissimilar to the ones in the learning set.

An other limitation of active shape models concerns their inability to statistically capture image features around shape models. Cootes et al. [24] solve this problem by taking samples of the image intensity perpendicularly to a shape. By recording intensity profiles for all labeled points of the shape model, they arrive at an augmented learning set. As with the shape data, they compute the statistics of the intensity data using equations 3.1-3.6. The image features improve model specificity and hence segmentation accuracy. Other work on modeling image features includes Van Ginneken et al. [122], who propose a selection of optimal features by non-linear classifiers. We adopt the idea of modeling both shape and image features.

In [24], it is assumed that the image feature values sampled at different points
along the boundary are independent of one another while commonly feature values are spatially highly correlated. Hence, independence of image feature values along a boundary cannot always be assumed. For this reason, Haslam et al. [56] propose a probabilistic fitness measure using concatenated intensity profiles, bringing in some continuity in feature values. This way, in addition two resolving the issue of dependence, they reduce the number of point distributions to two. Other work acknowledging and adhering to the often continuous characteristic of boundary features, particularly shape features, is found in e.g. [68], [27], [5], [56], [101], [71]. We embrace the thought of capturing spatial correlations among image features and we extend this idea to arbitrary features.

To overcome the problem of high correlation between structurally different features [56], Edwards et al. [37], and more recently Cootes et al. [19], propose an active appearance model that couples shape and image models more explicitly by using a single vector containing both shape and image feature values. By concatenating the feature values into a single vector and representing this vector as a point in a very high-dimensional space, a learning cluster is formed of combined shape and image features. This way a more specific boundary model is obtained and more accurate image segmentations are achieved. We adopt and generalize the idea of capturing correlations between structurally different boundary features.

In conclusion, many of the problems addressed by the aforementioned methods emanate from the discrete point representation of boundaries. Discrete points lead to discretization problems when point sets have different numbers of elements, requiring pseudo-continuous solutions. In addition, in discrete point representations, spatial interdependencies between features and correlations among structurally different features are not reckoned appropriately. Also, salient features in the learning set are insufficiently exploited when specified a priori rather than selected from the data by optimality. Finally, information valuable for model specificity is lost in the currently applied procedures for dimensionality reduction.

It is the purpose of this study to address these problems integrally. We do this with a technique we call strings.

### 3.3 Strings by Functional Data Analysis

Object boundaries are continuous. Hence, they should be represented by curves rather than by a collection of discrete points. Apart from this, boundaries are often described by multiple features. Hence they should be represented by multivariate models rather than by a combinations of univariate models. These boundary characteristics motivate the representation of boundary features by multi-variate curves in functional space, rather than by points in vector space. When boundaries are represented this way, the learning and segmentation problems can be solved by functional data analysis [95] (see figure 3.1).

Each of the components in figure 3.1 will be described in more detail in the following subsections. In section 3.4 we will again elaborate on these components one
Learning Phase

\[ \begin{align*}
I_m(x), s_m(v), & \quad i < M \quad (1\ \text{time}) \\
I_0(x), & \quad s_0(v)
\end{align*} \]

Segmentation Phase

\[ s_i(v) \quad (T\ \text{times}) \]

Figure 3.1: Overview of the string segmentation technique. Note that at the start of the learning phase, we need a set of images \( I_m(x), m = 1...M \) with corresponding known segmentations \( s_m(v) \).

Invariably, in this chapter bold face upper case indicates a matrix of functions, e.g. \( A(v), B(v), E(v) \) or in the case of \( G \) of scalars, bold face lower case indicates a vector of functions, e.g. \( f_t \), or scalars, e.g. \( g_i \), and regular lower case indicates a function or a scalar.

by one to show the (intermediate) results for a vertebra application.

3.3.1 Feature Function Definition

In order to construct a statistical boundary model, example features need to be specified and their values computed. Commonly, the feature values are a set of coordinate values of some boundary points and a set of image gradient values recorded orthogonally to those boundary points, see for example [20], [37], [19], [35].

We explore boundary features in example images using their known segmentation, represented by smooth curves \( s : \mathbb{R} \rightarrow \mathbb{R}_2 \) parameterized by \( v \in \mathbb{R} \). Given the set of \( M \) input images \( I_1(x), ... I_M(x) \) with corresponding segmentations \( s_1(v), ... s_M(v) \), the learning set consists of pairs of image and shape data (compare to equation 3.2)

\[ \mathcal{L} = \{(I_1(x), s_1(v)), ..., (I_M(x), s_M(v))\}. \quad (3.7) \]

For the \( m \)th learning example, the shape \( s_m(v) \) relates to the image at points \( I_m(s_m(v)) \). The relation is expressed in terms of \( N \) features derived from the shape (e.g. curvature) as well as from the image (e.g. isophote curvature). The mapping \( f : \mathbb{R} \rightarrow \mathcal{F} \) takes care of this, yielding feature functions \( f_m^{**}(v) \) in the \( N \)-dimensional functional space \( \mathcal{F} \), where each dimension corresponds to one feature, i.e.

\[ f_m^{**}(v) = [f_{m1}(v), ..., f_{mN}(v)]. \quad (3.8) \]
To capture image and shape features on and off a boundary we exploit the local Taylor expansion [64] of the image up to the second order, sampled at discrete but dense points along the boundary. In this way, we have an approximately complete two-dimensional description of local boundary properties. Later, the principal components analysis, yet to be described, will cancel out linear dependencies in the Taylor set.

### 3.3.2 Feature Function Alignment

For statistical analysis of example boundaries, feature values at one point on an example boundary need to be compared with values at an equivalent point on an other example boundaries. Commonly, this is achieved by scaling, rotating and translating the examples so that they correspond as closely possible. This reduces to aligning a set of discrete points, when boundaries are represented by point distribution models as in [20], [37], [19].

In the context of our functional data, the alignment problem is a curve registration problem. Feature functions \( f^v_m \) may differ due to the fact that they are not measured at the same path position \( v \) or due to small non-linear differences. A shift of feature values along \( v \) and a non-linear warping account for this. Alignment reduces to finding the warping function \( \omega_m(v) \) that produces the warped feature function

\[
f^v_m = f^v_m(\omega_m(v)).
\]  

(3.9)

The warping function \( \omega_m(v) \) is strictly monotonic and differentiable up to a certain order. It takes care of a shift and a non-linear transformation by the roughness penalty approach described in [95]. In this case, we penalize by the size of the third derivative of \( \omega_m(v) \).

Alignment of \( f^v_m \) is done by the Procrustes method [50] using a global alignment criteria that computes the least squares distance to \( \hat{f}(v) \), the overall average feature function. This reduces to finding \( \omega_m(v) \) such that

\[
\omega_m(v) = \arg\min_{\omega^v_m} \sum_{m=1}^M \int_v ||f^v_m(\omega^v_m(v)) - \hat{f}(v)||^2 dv.
\]  

(3.10)

The warping functions are estimated in an iterative process where argument values for a feature function are shifted and transformed so as to minimize the least squares error. The estimated average \( \hat{f}(v) \) is updated by re-estimating it from the partially aligned feature functions.

The final average feature function \( \bar{f}(v) \) is computed from the aligned set. It is subtracted from each feature function to normalize the range of feature values. This yields

\[
f_m(v) = \frac{f^v_m(v) - \bar{f}(v)}{\sigma_f(v)}
\]  

(3.11)

with units of variance due to normalization by the variance vector of functions

\[
\sigma_f(v) = \left( \frac{1}{M} \sum_{m=1}^M ||f^v_m(v) - \bar{f}(v)||^2 \right)^{1/2}
\]  

(3.12)
Normalization is required to reduce the influence of differences in variation of differently measured features. The aligned normalized feature functions $f_m(v)$ contain all information needed to statistically summarize features into a boundary model.

### 3.3.3 Feature Space Reduction

We perform principal components analysis to project the high dimensional functional data to a smaller feature space expecting that the essential structure in the original data is preserved. This is admissible as long as the features exhibit a small number of modes of variation, covering a large part of the variability in the data.

Functional principal component analysis [31] computes the main modes of variation in the collection of $N$-dimensional feature functions $f_m(v)$. The number of modes to retain is derived from a given proportion of the variance as explained in the learning set. When the modes are numbered by $q = 1, \ldots, Q$, the central concept is that of taking the linear combination

$$g_{mq} = \sum_{n=1}^{N} f_{mn}(v) \alpha_{qn}(v) dv,$$

where $\alpha_{qn}(v)$ denotes a weighting function chosen so as to highlight variation in the data in dimension $n$. As before, $f_{mn}(v)$ is the $n$th dimension of the $m$th observed feature function. The values $g_{mq}$ are the principal component scores. They will be used to produce more robust descriptions [32].

To obtain the value of $g_{mq}$ for all $q$ the corresponding vectors of weighting functions $\alpha_q(v) = [\alpha_{q1}(v), \ldots, \alpha_{QN}(v)]$ need to be computed. They are sought for one-by-one in such a way that they explain most of the variation in the learning data

$$\alpha_q(v) = \arg \max_{\alpha_q(v)} \frac{1}{M} \sum_{m=1}^{M} g_{mq}^2 \quad \text{subject to} \quad \sum_{n=1}^{N} \int_v f_{mn}(v) \alpha_{qn}(v) dv = 1 \quad (3.13)$$

where $\alpha_k(v)$, for each iteration $k$, is subject to the following orthonormal constraints

$$\sum_{n=1}^{N} \int_v \alpha_{qn}(v)^2 dv = 1 \quad (3.15)$$

$$\sum_{n=1}^{N} \int_v \alpha_{kn}(v) \alpha_{qn}(v) dv = 0, k \leq q. \quad (3.16)$$

Equations 3.15 and 3.16 ensure that the vector $\alpha_1(v)$ contains most of the independent variation. After $\alpha_1(v)$ has been established, the above process is continued until all $Q$ significant modes of variation, each described by $\alpha_q(v)$, are obtained.

The matrix of functions $A(v) = [\alpha_1(v), \ldots, \alpha_Q(v)]^T$ indicates where along the boundary there is independent variation in the learning ensemble. $\alpha_1(v)$ captures the
location with largest correlated variation, \( \alpha_2(v) \) the second largest of the remaining variance and so on until most of the variation is explained. Hence, the functional principal components capture the most important feature subspace by the matrix of weighting functions \( A(v) \).

### 3.3.4 Principal Components Regression

To construct an underlying model of feature values, the distribution of feature values seen in the learning set needs to be captured in statistical terms. In the active shape model literature [20], [37], [19], [35], feature reconstruction is done after projecting the features to the space spanned by the most important principal components. Hence, only part of the data in the learning set is subjected to modeling. We perform principal component regression [14] to obtain a predictive model from the feature functions in the learning set.

We define the matrix of functions \( F(v) = [f_1(v), ..., f_M(v)]^T \) and the matrix \( G \) by

\[
G = F(v)A(v)
\]

with scalar elements according to the dot product defined in equation 3.13. We define a matrix of regression functions \( B(v) = [\beta_1(v), ..., \beta_Q(v)]^T \), with elements of the same N-dimensional functional form as the elements of \( F(v) \). To find the values of \( B(v) \), the matrix of feature functions \( F(v) \) is expressed as

\[
F(v) = GB(v) + E(v)
\]

with \( E(v) = [\varepsilon_1(v), ..., \varepsilon_M(v)]^T \) being the matrix of residual functions yet to be defined. Instead of regression on the original feature data, regression is performed on the principal component scores containing information on how the feature samples correlate with one another. The matrix of regression functions \( B(v) \) gives an estimate of how the principal scores relate to the feature functions and what the contribution of each is towards defining an unknown feature function. The regression functions are computed by least squares minimization such that

\[
B(v) = \arg\min_{B^*(v)} \sum_{m=1}^{M} \int_v ||f_m(v) - g_mB^*(v)||^2 dv.
\]

Since there are no particular restrictions on the way in which the matrix of functions \( B(v) \) varies as a function of \( v \), the solution can be obtained by minimizing the least squares difference for each \( v \) separately. After least squares minimization we have

\[
E(v) = ||F(v) - GB(v)||^2.
\]

With help of the estimated regression functions we predict scores for an unknown feature function in the segmentation phase, reconstruct it in reduced space according the regression model and evaluate it by examining the distance of its score to the cluster of scores corresponding to the feature functions in the learning set.
3.3.5 Mahalanobis Distance

In order to determine how a single boundary relates to the collection of boundaries in the learning set, a distance measure needs to be defined. In the active shape model literature, the evaluation of boundary feature values is performed in terms of the Mahalanobis distance, i.e. the distance to the average normalized by the variation in each dimension. Following Cootes et al. [20], we use a Mahalanobis distance model [32] to compute the distance of a feature function to the average of the learning set.

The Mahalanobis distance model is obtained by augmenting $G$ with the vector of residuals $\epsilon = [\epsilon_1, ..., \epsilon_M]$ with elements defined as

$$
\epsilon_m = \frac{1}{N} \sum_{n=1}^{N} \left( \int_v \epsilon_{mn}(v) dv - \frac{1}{M} \sum_{m=1}^{M} \int_v \epsilon_{mn}(v) dv \right).
$$

This yields an $(Q + 1) \times M$ augmented principal components scores matrix $G^* = [G, \epsilon^T]$. This improves the robustness of Mahalanobis distance calculation [32]. The Mahalanobis distance matrix is then defined as

$$
D = \frac{G^{*T}G^*}{(M + 1)}.
$$

The Mahalanobis distance of $f_m(v)$ to the average $\bar{f}(v)$ is computed using $g^*_m = [g_m, \epsilon_m]$ as follows

$$
D^2(f_m(v), \bar{f}(v)) = g^*_m D^{-1} g^*_m T.
$$

Note that $D$ depends on $g^*_m$. Hence, the distance model not only takes into account variations encountered in the learning set but also the valuable additional discriminating factor of residual information [69]. In the segmentation phase, the Mahalanobis distance model will be used as an objective function to find a boundary in a new image from which a feature function emanates similar to the ones seen in the learning set.

3.3.6 Stochastic Outlier Removal

As outlier feature functions in the learning set can have a severe influence on the discrimination ability of the Mahalanobis distance model, they are removed following [35]. Those with a Mahalanobis distance exceeding a threshold $\tau_M$ are considered outliers. For a Gaussian distribution, $\tau_M = 3$ corresponds to removal of all instances that have a 1% probability of belonging to the class. These instances are removed, resulting in a new learning set in which each element has a Mahalanobis distance of $\tau_M$ or less. Learning is done once again to come up with more appropriate regression and distance models. For simplicity of notation, we assume in the following that the quantities are computed from the reduced set rather than from the full set of feature functions used this far.
3.3.7 Deformable Strings

Having constructed a statistical boundary model in the learning phase, in the segmentation phase we use this model as a reference for finding a boundary in a new image. To this end, we use the following information from the learning phase: the average feature function \( \tilde{f}(v) \), the normalization function \( \sigma_f(v) \), the Mahalanobis distance model \( G^* \) and the regression functions matrix \( B(v) \).

Now, consider the active feature function \( f_t^*(v) \), which will be deformed in time \( t \). We call it a string. The string lives in the \( N \)-dimensional feature space \( \mathcal{F} \), and hence we can compute its Mahalanobis distance to \( \tilde{f}(v) \). As in the learning phase (see equation 3.9), we first obtain

\[
f_t^*(v) = f_t^{**}(\omega_t(v)),
\]

where, similar to equation to 3.11, the warping function \( \omega_t(v) \) is computed such that

\[
\omega_t(v) = \arg\min_{\omega_t(v)} \int_v \|f_t^{**}(\omega_t(v)) - \tilde{f}(v)\|^2 dv.
\]

We normalize the feature function in analogy to equations 3.11 and 3.12. Normalization yields

\[
f_t(v) = \frac{f_t^*(v) - \tilde{f}(v)}{\sigma_f(v)}.
\]

The quality of \( f_t(v) \) with respect to the reduced feature functions matrix \( F(v) \) is determined from the relation of its corresponding score vector \( g_t \) to the cluster of example scores contained in \( \mathbf{G} \). The vector \( g_t \) is estimated by solving

\[
f_t(v) = g_t B(v)^T + \epsilon_t(v).
\]

That is, the equation estimates the score of a new feature function on the basis of the principal component regression model obtained in section 3.4. The principal component scores are estimated by least squares minimization such that

\[
g_t = \arg\min_{g_t} \int_v \|g_t B(v)^T - f_t(v)\|^2 dv.
\]

In analogy to equation 3.20, we augment \( g_t \) with the residual of the least squares minimization as an additional discriminating factor for the string. The \((Q+1)\)-vector \( g_t^* = [g_t, \epsilon_t] \) is obtained on the basis of

\[
\epsilon_t(v) = \|g_t B(v)^T - f_t(v)\|^2
\]

by adding to \( g_t \) the residual sum

\[
\epsilon_t = \frac{1}{N} \sum_{n=1}^{N} \left( \int_v \epsilon_{tn}(v) dv - \frac{1}{M} \sum_{m=1}^{M} \int_v \epsilon_{mn}(v) dv \right).
\]
The Mahalanobis distance of \( f_i^{**}(v) \) is defined as the distance of the newly augmented score \( g_i^* \) to the average of the cluster of example scores

\[
D^2(f_i(v), \bar{f}(v)) = g_i^*D^{-1}g_i^T.
\]  

(3.31)

We use this quality measure for image segmentation by strings. The string \( f_i^{**}(v) \) is defined by features extracted from an active shape model \( s_t(v) \) and from the image in which that shape model lives. The string vibrates in feature space due to deformations of the shape model in image space. In this iterative procedure, the shape model is freely deformed rather than constructed in the reduced space as is the case in [20] and [37]. Forms of the shape model that are less plausible are punished by rather than prohibited.

### 3.3.8 Optimization

Finally, we formulate the segmentation problem as an optimization problem. The objective is to find a boundary that gives rise to a feature function with minimal Mahalanobis distance \( D(.) \) to the ones seen in the learning set. To this end, the deformable shape model \( s_t(v) \) is deformed in the image \( I_0(x) \) to suggest a statistically optimal boundary described by the deformable string \( f_i^{**}(v) \). Starting from an initial shape configuration \( s_{t=0}(v) \) the shape model is deformed by tuning its shape parameters in such a way that the state of minimal energy provides the optimal feature function. This reduces to optimizing \( s_t(v) \) such that

\[
f(v) = \arg\min_{f_t(v)} D(f_t(v), \bar{f}(v)).
\]  

(3.32)

We use simulated annealing [63] for optimization as it distinguishes between different local minima in the energy landscape. Starting off at the initial configuration, a sequence of iterations is generated, where each iteration consists of the random selection of a configuration from the neighborhood of the current configuration and the calculation of the corresponding change in the energy value. By a small perturbation in the neighborhood, a transition is achieved from one configuration into another one. If the change from time \( t \) to \( t + 1 \) yields negative \( \Delta D = D(f_{t+1}(v), \bar{f}(v)) - D(f_t(v), \bar{f}(v)) \), the transition is accepted unconditionally; if the cost function increases the transition is accepted with a probability based upon the Boltzmann distribution \( p = e^{-\frac{\Delta D}{kT}} \) where \( k \) is a constant and the temperature \( T \) is a control parameter. This temperature is gradually lowered throughout the segmentation from a sufficiently high starting value, i.e. a temperature where almost every proposed transition, both positive and negative, is accepted to a freezing temperature, where no further changes occur. The temperature is decreased in stages, and at each stage the temperature is kept constant until thermal quasi-equilibrium is reached.

### 3.4 Experiments and Results

We illustrate and discuss the use of functional data analysis for construction of the boundary model and the application of strings for image segmentation. We do this...
step by step according to figure 3.1. To this end, we use 145 annotated and digitized NHANES X-ray images of normal cervical vertebrae, acquired from the National Center for Health Statistics (NCHS) [86]. As can be seen from figure 3.8, the vertebra boundary in these images is ill-defined. It is characterized by the presence of interfering boundaries (e.g. the vertebra above and below), convoluted boundary parts (e.g. tips of the vertebral body), missing image evidence (e.g. at the pedicles) and in this case very poor image quality. These characteristics complicate model construction and image segmentation.

3.4.1 Feature Instantiation

For the description of the boundary we use a repertoire of features. Among the many features that have been proposed in literature (see [44] and [73] for a comprehensive survey of image and shape feature respectively), we confine ourselves to the use of invariant features. Invariant features generalize applicability, but more importantly, they minimize the need for feature alignment. For this reason, we use the features listed in table 3.1.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Feature</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>contour curvature [44]</td>
<td>[ s_x(v)z_y(v) + s_y(v)z_x(v) ]</td>
</tr>
<tr>
<td>2</td>
<td>isophote curvature [129]</td>
<td>[ I_{xx}(x)I_{xy}(x) - 2I_{x}(x)I_{xy}(x) + I_{yy}(x)I_{x}(x)^2 ]</td>
</tr>
<tr>
<td>3</td>
<td>directional correspondence [79]</td>
<td>[ \nabla I(s(v)) \cdot n(v) ]</td>
</tr>
</tbody>
</table>

Table 3.1: Features in our implementation defining the dimensions of \( f^{**}(v) \). The first dimension is the contour curvature, the second the isophote curvature and the third the directional correspondence between the normal \( n(v) \) to the shape \( s \) at \( v \) and the image gradient at \( \nabla I(x) \) with \( x = s(v) \).

The features listed in table 3.1 and their derivatives up to second order are used. As a consequence \( N = 9 \) features are measured along each example vertebra boundary. As in this application the example vertebra boundary is represented by 7 discrete points, manually marked in the image by a medical expert, we compute a continuous approximation of the boundary by interpolation of a curve through the 7 points. The curve \( s(v) \) and its corresponding 9-dimensional feature function \( f^{**}(v) \) are represented by splines. They are formulated as tensor product B-Spline curves [93]

\[
s(v; p) = \sum_{k=1}^{K} B_{k}(v) p_{k} \quad (3.33)
\]

\[
f^{**}(v; q) = \sum_{j=1}^{J} B_{j}(v) q_{j} \quad (3.34)
\]

Basis functions \( B_{k}(v) \) correspond to the \( K = 7 \) manually marked points \( p_{k} \). The \( N = 9 \) features, and also point coordinates values, are computed along \( s(v; p_{k}) \) at
100 samples. The 100 coordinate values are used for redefinition of the 7-vector of control points \( \mathbf{p}_k \) to a 100-vector of control points, to be used in active shape model segmentation (yet to be described). Feature functions \( \mathbf{f}^{**}(v; \mathbf{q}_j) \) are defined by basis functions \( B_j(v) \), corresponding to the \( J = 100 \) N-dimensional control points \( \mathbf{q}_j \), i.e. the sampled feature values. As these feature values are not always smooth we impose regularity by using basis expansions with a relatively small number of basis functions [95].

### 3.4.2 Learning Phase

We discuss the results of the learning phase for the vertebra application. We only discuss the 0th-order derivative values of the features, i.e. only the first three dimensions of \( \mathcal{F} \).

![Figure 3.2: Feature values along path parameter \( v \) for all \( M = 145 \) examples. Left: contour curvature values \( f^{**}_{m1}(v) \). Middle: isophote curvature values \( f^{**}_{m2}(v) \). Right: directional correspondence values \( f^{**}_{m3}(v) \).](image)

Figure 3.2 illustrates contour curvature values \( f^{**}_{m1}(v) \), isophote curvature values \( f^{**}_{m2}(v) \) and directional correspondence values \( f^{**}_{m3}(v) \) computed from \( s_m(v) \) and the image data arrays (see equation 3.8). The presence of correlated structure in the contour curvature values is apparent. The peaks in curvature values correspond with the tips of the vertebral body and correlate with the peaks in \( f^{**}_{m2}(v) \). This is more clearly seen from the average functions in figure 3.5. Note that the isophote curvature is badly defined along most parts of the vertebra boundary. More correlated structure is seen in the feature values of directional correspondence.

The results of aligning feature functions \( f^{**}_{m}(v) \) by the iterative Procrustes procedure in equation 3.9 is illustrated in figure 3.3. In the vertebra application, the majority of the feature functions already has a good alignment thanks to the a priori manual registration of the discrete points by medical experts (from which the continuous shapes and feature functions are computed). As a consequence, alignment brings no significant changes to feature functions \( f^{**}_{m}(v) \). Alignment is more important during segmentation where the starting points for sampling features are unknown.

The effects of normalization of feature functions \( f^{*}_{m}(v) \) are shown in figure 3.4. Note that both the isophote curvature and directional correspondence show feature
functions with large variation, amplified due to centering to zero mean and unit variance according to equation 3.11. The feature functions with extreme variation are candidate outliers, to be removed later from the learning set.

The composite effect of adding and subtracting two standard deviations of the first principal component \( \alpha_1(v) \) to the average feature function \( \bar{f}_m(v) \) is shown in figure 3.5. We have chosen to reduce the feature space using \( Q = 4 \) principal components (see equation 3.14), together capturing 83.9 percent of the total variability in the data. \( Q \) has been set to 4 because we expect for our application that there are 4 corners, and hence places, in the model where the data in the learning phase show there is independent variation in the boundary feature values. Note that these displays remind of the diagrams in physics of modes of vibration in a string fixed at both ends, hence the name strings.

Finally, regression functions \( \mathbf{B}(v) \), obtained using equation 3.19, are shown in figure 3.6. The function \( \beta_{qn}(v) \) indicates how the \( n \)th feature along the curve contributes to the \( q \)th principal component. Hence, the regression functions indicate which features are locally most important to define the boundary characteristics, implying that...
Figure 3.5: Average feature functions and the effects of adding (+) and subtracting (-) two standard deviations of the first principal component: $\tilde{f}(v) \pm 2\text{ standard deviations of } \alpha_1(v)$. From left to right the effects for, contour curvature, isophote curvature and directional correspondence.

Figure 3.6: Regression functions corresponding to the $Q = 4$ principal components. Left: functions $\beta_{11}(v), ..., \beta_{41}(v)$ for contour curvature. Middle: $\beta_{12}(v), ..., \beta_{42}(v)$ for isophote curvature. Right: $\beta_{13}(v), ..., \beta_{43}(v)$ for directional correspondence.

weighting is done in a way that exploits the most correlated and descriptive features at each boundary point.

Not shown in illustrations is the effect of the optional outlier removal. A total of 11 example feature functions have been considered outliers and removed. After this, the whole learning procedure is repeated one more time. The average feature function vector $\tilde{f}(v)$, the normalization function vector $\sigma_f(v)$, the matrix of regression functions $B(v)$ and of scalars $G^*$ are transferred to the segmentation phase.

3.4.3 Segmentation Phase

We perform string segmentation of the vertebra images using the above learning results as a reference. An active B-spline shape $s_t(v; p_k)$, defined by $k = 100$ control points, is deformed in the image by repositioning of these points in an iterative procedure. Each time a total of 100 samples are taken along $s_t(v; p_k)$ to construct a feature function from them. For simplification of comparison with active shape model seg-
mentation (described below), the samples are taken at points $p_k$. For computation of image features, derivatives are computed by convolution of the image with Gaussian derivatives at scale $4.0$. Optimization of the active shape model is done by the simulated annealing optimization procedure, with the Boltzmann factor set to $0.99$, thermal equilibrium defined as a 10 percent or smaller change in 10 random trials and a maximum of 50 iterations.

We also perform multi-resolution active shape model segmentation [23]. The active shape model learns the distribution of points $p_k$. The normalized gradient is captured perpendicularly to the boundary in profiles of length $3$. The number of levels of resolution is set to $6$, with level $0$ the original image, level $2$ the image with half the number of pixels etc. After alignment of the shapes by rotation, translation and scaling, a shape model is constructed in which $84.2$ percent of the variance is explained by the first $6$ principal components. The neighborhood examined to find a better location for each point is $9$. The initial shape is not taken to be the statistical average, rather it is an arbitrary shape which is projected onto the space spanned by the $6$ principal components for evaluation [53]. Optimization stops after a maximum of 50 iterations.

We perform three experiments with strings and active shape models (software courtesy of [53]). In each experiment a shape is placed in the image on the correct position and perturbed a known amount to verify robustness against initialization. The perturbations include translation up to $15$ pixels, rotation up to $30$ degrees or scaling with respect to a center point $c$ with a factor up to $0.1$. The perturbed shape is then used to bootstrap segmentation. To measure the accuracy of the segmentation the distance from the resultant to the correct boundary is computed using the root squared metric error. The error measure is based on the $K = 100$ optimized control points and on control points $p_k$ defining ground-truth, also by 100 samples. The experiments are performed systematically excluding each learning instance, from the learning set and using the excluded one to test the performance of the model built without it.

Figure 3.7 shows the average root squared metric of 145 segmentations for varying amounts of translation, rotation and scaling. As can be seen from the reduction in the root mean squared error, the initial shape almost always moves to the correct boundary for both strings and active shape models. The string segmentation method outperforms active shape model segmentation when the initial shape is close to the correct boundary, in spite of the fact that the ground-truth is poorly sampled. If not initialized close to the target, string segmentation produces results similar to those produced by active shape model segmentation, occasionally worse. This sensitivity to initialization is attributed to the fact that no pose parameters are optimized to explicitly account for pose corrections, in contrast to active shape model segmentation. The active shape model finds the correct boundary even from a large distance if that boundary is well defined. However, when the image evidence is vague, the active shape model tends to get trapped in a local minima far away from the correct boundary due to wrong pose optimization. For strings, we expect performance improvement by optimization of pose parameters in addition to shape parameters under the condition that pose parameters are restricted to admissible ranges. Equivalent improvements
may be expected for active shape models by restricting pose parameters.

**Figure 3.7:** The average root mean square distance for initial and final curves as a function of translation (left), rotation (middle) and scaling (right).

From figure 3.7 we observe that even when the initial shape is the correct shape, optimization brings the initial shape to rest at an average of almost 5 pixel distance from correct shape. We note that the points marking the vertebra boundaries in our learning set have been placed by a single medical expert and that variation of 5 or more pixels in manual point placement can be expected. Hence, the structural error of approximately 5 pixels is largely ascribed to intra-observer variability. We expect an improvement of performance proportional to the accuracy of the ground-truth segmentation, either by more precise individual assignment or by using larger amounts of salient points per vertebra. As the construction of a general model for the cervical vertebra also contributes to the structural error, we expect that a dedicated boundary model, e.g. a model of the C1 vertebra, will bring in more accuracy and specificity to the boundary model.

**Figure 3.8:** Segmentation of a NHANES cervical vertebra image. Left: typical NHANES image with ground-truth delineation. Middle: active shape model segmentation (solid line). Right: string segmentation (solid line). Note the vague image evidence along the vertebra boundary due to the very low image quality.

Figure 3.8 shows an example result for the active shape model segmentation and
the string segmentation, which typically takes about 15 seconds and 95 seconds respectively on a standard machine. In spite of the elaborate matrix function manipulations, performance is still very good. Much of the processing time is attributed to feature function alignment. Also the exhaustive search of simulated annealing contributes to the high computational cost. We expect a significant reduction in computational cost with landmark-based alignment, a more efficient optimization technique and optimization of our code for speed. Note that much of the erroneous solutions of the active shape model [53] in figure 3.8 are global of nature. Apparently, the image evidence around the target boundary is too vague to be conclusive for such a model. Hence, the active shape model converges with a feasible shape in an unacceptable pose. For the string segmentation, the erroneous solutions are confined to boundary segments not on the correct boundary, but rather on other visually-detectable edges, such as tissue/background edges or even edges produced by gray-scale intensity variation within a vertebra.

### 3.5 Discussion and Conclusion

In conclusion, we have addressed a number of contemporary problems with statistical image segmentation models. We have represented boundaries by curves, solving some problems arising from discrete representations. By placing these curves in a multi-variate functional feature space we have properly dealt with the problem of spatial and feature interdependencies. We have performed curve registration, releasing us from the problem of missing points when aligning. We have substituted feature data by principal component scores, thereby not only reducing the feature space but also endorsing more robust computations. We have constructed a regression model for predicting unknown boundaries, even ones dissimilar to the examples. We have built distance model that also accounts for residual information. Finally, we have defined image segmentation as string optimization in multi-dimensional feature space.

In this paper we have applied the string method for segmentation of rigidly shaped objects with ill-defined image evidence along their boundary, as often found in medical images. The method is also expected to work well for objects with more articulated shapes, provided there is correlated variation in the shape properties. When applied to objects of the real world, where objects often have an unknown and varying background, the definition of features should be such that values are computed from the inside of the object only. This provides a multi-variate statistical description of the object area, rather than a description of the object boundary with its surrounding. It is important to note that continuous ground-truth segmentations are required to fully exploit the capabilities of strings.

With [20], [37], [19], [35] we share the observation that features should be learned rather than constructed from a priori geometrical or analytical knowledge. Geometrical and analytical features such as smoothness act as constraints on the solution since the resulting shape then will often be smooth at most places, regardless whether that is the appropriate solution or not. As in [20], we conceive of shape and image evidence as features and learn where they are effective in describing the statistics of the model.
In fact, we have adopted the idea of mapping features to a space where the most important modes of variation are determined by principal components analysis and used for steering a shape for segmentation of an image.

The difference is that we conceive of a boundary as a multivariate continuous curve, requiring continuous functions to be learned. Cootes et al. [20] reduce the boundary to point sets. This introduces the following problems. First, points may be confused with other points if they are not labeled, leading to erroneous classification. Second, point location correspondence may be doted with error. Third, as recognized by Dutta [35], points may be missing, requiring a pseudo continuous reconstruction of the boundary. Finally, measurements at discrete sample points are less reliable and lead to lose of spatial coherence.

We are less critically effected by these difficulties as we use closed continuous curves. We profit from functional data analysis in exploiting the spatial and feature correlations to explain the observed variations in the training data rather than removing or down weighting such correlations. Apart from this, in explaining these variations, we also consider residual information, rather than omitting it. Contrary to Cootes et al. [20] we built a regression model from which we try to explain unknown boundaries, including the errors made in doing so. Another difference is that, instead of producing new models in reduced space, and hence restricting the shapes the method can handle, we freely produce new segmentation shapes and punish implausible shapes.

In comparing the performance of our string implementation with an active shape model implementation using software by [53], we note the following. The active shape model performs better in the case the target object is visually well-defined and the initial shape is placed at a large distance from the target. This is thanks to the explicit optimization of pose parameters for translation, rotation and scaling. When the initial shape is initialized close to the target object, the active shape model performs better in the case the object has boundary properties like most objects in the learning set. If the target object has boundary properties that are dissimilar to the ones in the learning set, the string model produces more accurate results. We note that the prediction of feature values in reduced space, aids in explaining objects not seen in the learning set.

In a situation where the target boundary is visually ill-defined and the shape initialization is at a large distance from the target, both string segmentation and active shape model segmentation perform poorly. For that condition, we found active shape models to produce much better or much worse results than string segmentation. On average their performance is then similar. It has been acknowledged previously that proper initialization is required to guarantee satisfying result in view of the presence of disturbing attractors in the image [83]. Strings outperform active shape models if the target object is visually ill-defined and the initialization is close to the target boundary. This is due to the fact that strings locally exploit the discriminative power of a repertoire of features. Also the fact that strings cope better with spatial dependencies between feature values and the fact that they take into account correlations among features more explicitly improves segmentation under such highly demanding conditions.
Hence, we arrive at the conclusion that strings are particularly suited for learning variational models of objects that have inherently multivariate continuous boundary characteristics. Strings are also very well suited for segmentation of complex scenes, where the visual evidence is vague or where a multi-dimensional feature set is needed to capture an object boundary.