In: Proceedings of the International Conference on Machine Learning (ICML '07), Oregon, USA (2007).

Linear and Nonlinear Generative Probabilistic Class Models for Shape Contours

Graham McNeill GRAHAM.MCNEILL@ED.AC.UK
Sethu Vijayakumar SETHU.VIJAYAKUMAR@ED.AC.UK
Institute of Perception, Action and Behavior, University of Edinburgh, Edinburgh EH9 3JZ, UK.

Abstract

We introduce a robust probabilistic approach to modeling shape contours based on a low-dimensional, nonlinear latent variable model. In contrast to existing techniques that use objective functions in data space without explicit noise models, we are able to extract complex shape variation from noisy data. Most approaches to learning shape models slide observed data points around fixed contours and hence, require a correctly labeled 'reference shape' to prevent degenerate solutions. In our method, unobserved curves are reparameterized to explain the fixed data points, so this problem does not arise. The proposed algorithms are suitable for use with arbitrary basis functions and are applicable to both open and closed shapes; their effectiveness is demonstrated through illustrative examples, quantitative assessment on benchmark data sets and a visualization task.

1. Introduction

Statistical shape models of 2D contours are used in medical image analysis, object recognition and image retrieval. To learn an accurate model of shape variability one must either know the correct correspondence between all shapes of the training set *a priori*, or learn the correspondences and the model simultaneously; in this paper, we consider the latter situation (Figs. 1 and 2). Many of the previous approaches to this problem are conceptually similar (Kotcheff & Taylor, 1998; Davies et al., 2002; Ericsson & Astrom, 2003b; Thodberg & Olafsdottir, 2003; Hladuvka & Buhler, 2005): given some observed points from multiple contours, fit curves (often polylines) to the data and then slide points around each curve to find the optimal correspondence with respect to an objective function. Noise in the data is usually not modeled and a 'reference shape' with fixed points/parameterization is typically required to prevent degenerate solutions whereby points cluster about a single region of the contour.

The method proposed by Kotcheff and Taylor (1998) forms the basis of much recent work in this area. Given a Procrustes alignment of the training shapes, they monotonically reparameterize each training shape so as to minimize the determinant of the sample covariance matrix. A Gaussian noise model with *prespecified variance* is introduced to avoid numerical problems. Davies et al. (2002) use a similar formulation, assuming a Gaussian model over the training shapes and using the minimum description length (MDL) framework to learn smooth reparameterization functions (RFs). MDL is typically used for model selection whereby model complexity is balanced against the ability of the model to explain the data. However, in the approach of Davies et al. (2002) the model is fixed and the objective function is similar to that used by Kotcheff and Taylor (1998).

A number of modifications to the MDL approach have been proposed. Thodberg has incorporated curvature information (Thodberg & Olafsdottir, 2003) and extended the MDL technique to appearance models (Thodberg, 2003). Efficient gradient-based techniques for learning RFs have been introduced for discrete (Ericsson & Astrom, 2003b) and continuous shape representations (Hladuvka & Buhler, 2005). Ericsson and Astrom (2003a) have proposed an alternative formulation to MDL which is invariant to affine transformations.

In this paper, we introduce the Probabilistic Contour Model (PCM) and the Nonlinear Probabilistic Contour Model (NPCM). The generative model for PCM/NPCM can be summarized as follows:

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

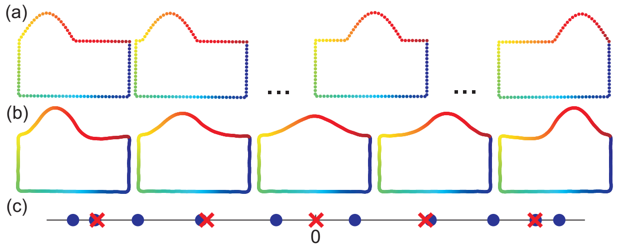

Figure 1. (a) 4 of the 10 point sets from the box-bump data set; color represents the value of the arc length parameter. (b) Curves associated with equal increments along the 1-dimensional latent space of a probabilistic contour model (PCM) that does not learn reparameterization functions (RFs). (c) Latent space: the circles are the posterior means of the 10 data shapes, the crosses are the latent variables of the curves in (b).

1. Sample a single point from a low-dimensional latent space.
2. Map the point to a 2D curve using a linear (PCM) or nonlinear (NPCM) mapping.
3. Reparameterize the curve, sample it and add Gaussian noise to the sample points to generate the observed data.

The inclusion of an explicit low-dimensional latent space and noise model contrasts with techniques which treat major and minor components of variation separately in the objective function (Davies et al., 2002; Ericsson & Astrom, 2003b; Thodberg & Olafsdottir, 2003; Hladuvka & Buhler, 2005). Neither the existing techniques nor PCM are designed to handle nonlinear patterns of variation and we extend PCM to NPCM to address this limitation. Since many data sets contain noise of *unknown* magnitude associated with either the shapes themselves (*e.g.* due to random factors in the biological processes that generated them) or the image capturing and processing techniques used for shape extraction, we estimate the noise variance during learning rather than specifying it *a priori*. In PCM/NPCM, the underlying (noise-free) contours are latent variables so, unlike other approaches, curves are not fitted to the data prior to learning the model. Also, the observed data points remain fixed rather than being slid around, so there is no danger of point clustering and hence, no need for a reference shape.

2. Shapes, Curves and Reparameterization Functions (RFs)

We start by introducing the main components that will be used to construct a shape model. An ordered 2D point set is represented by the matrix $\mathbf{X} \in \mathbb{R}^{J \times 2}$, where each row contains the coordinates of a single point and points are stacked in the order they appear on the underlying contour. We will often consider the long vector $\mathbf{x} \in \mathbb{R}^{2J}$ formed by concatenating the x-coordinates and the y-coordinates: $\mathbf{x} \equiv \mathrm{vec}(\mathbf{X})$.

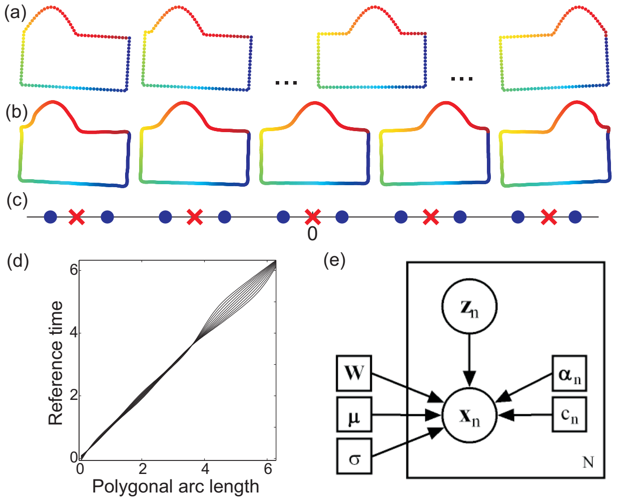

Figure 2. PCM for the bump-box data: (a) The reparameterized data shapes; color represents the value of the 'reference time' parameter. (b) Curves associated with equal increments along the latent space. (c) The latent space. (d) The RFs. (e) Graphical model for PCM.

A shape model is learnt from N ordered point sets, $\mathbf{X}_1, \ldots, \mathbf{X}_N$. To avoid complicating the notation, we assume that each point set contains J points; *point sets of unequal size are easily handled.* The 2D curve responsible for generating $\mathbf{X}_n$ is described by a linear combination of basis functions:

$$f_n(t) = \boldsymbol{\phi}^T(t)[\mathbf{v}_n^x, \mathbf{v}_n^y], \tag{1}$$

where t is the curve parameter, $\mathbf{v}_n^x, \mathbf{v}_n^y \in \mathbb{R}^K$ are vectors of coefficients and $\boldsymbol{\phi}(t) \equiv (\phi_1(t), \ldots, \phi_K(t))^T$ is a vector of basis functions. Again, we frequently consider the long vectors $\mathbf{v}_n \in \mathbb{R}^{2K}$ formed by concatenating $\mathbf{v}_n^x$ and $\mathbf{v}_n^y$. For closed curves, the basis functions are periodic, *e.g.* Fourier, wrapped Cauchy, von Mises, or periodic B-spline.

Given a training set, we assume that the *same point* (*e.g.* the top of the bump for the shapes in Fig. 1a) varies in both its spatial location and the distance along the shape perimeter at which it appears. The idea is to solve the correspondence problem by reparameterizing the curves from which the points were sampled, allowing us to construct a shape model over the spatial locations of corresponding points. This is demonstrated in Figs. 1 and 2 where an initial poor correspondence between training shapes (Fig. 1a – the color of the bump on the leftmost shape is noticeably different from that on the rightmost) is improved (Fig. 2a) leading to a better shape model – the generated shapes in Fig. 2b resemble the training shapes more than those in Fig. 1b. For further details of the box-bump experiment, refer to Sec. 6.1.

We now consider curve reparameterization in more detail and introduce a monotonic reparameterization function (RF), $g_n(t)$, for each curve, $f_n(t)$. RFs can be expressed as the integral of a linear combination of basis functions where both the basis functions and the coefficients are non-negative (Davies et al., 2002; Hladuvka & Buhler, 2005). To avoid these non-negativity constraints, we use a 'monotonicity operator' (Ramsay & Silverman, 2005): take an unconstrained linear combination of arbitrary basis functions, exponentiate to ensure positivity and then integrate to ensure monotonicity. Ramsay and Silverman (2005) have successfully applied this idea to curve registration using an intuitive least-squares error function. Here, registration is carried out with respect to the latent variable models introduced in the following sections.

Closed curves are more difficult to model than open curves since the start point on each shape is generally unknown and hence, each RF must contain a shift parameter. Focusing on the closed curve problem, we define the RF for the n-th shape, $g_n(t) : [0, 2\pi] \rightarrow [c_n, 2\pi + c_n]$ as

$$g_n(t) \equiv 2\pi \frac{\int_0^t \exp(\boldsymbol{\psi}^T(\tau)\boldsymbol{\alpha}_n)d\tau}{\int_0^{2\pi} \exp(\boldsymbol{\psi}^T(\tau)\boldsymbol{\alpha}_n)d\tau} + c_n, \qquad (2)$$

where $\boldsymbol{\psi} = (\psi_1(t), \ldots, \psi_Q(t))^T$ is a vector of basis functions and $\boldsymbol{\alpha}_n \in \mathbb{R}^Q$ is a vector of coefficients. The values $t = c_n$ and $t = 2\pi + c_n$ initially correspond to the same point on the curve so, assuming that smooth RFs are desired, we should ensure that the derivative of $g_n(t)$ is equal at c_n and $2\pi + c_n$. This is most easily achieved by using smooth 2π-periodic basis functions.

For each curve n, we approximate the value of the arc length parameter at each observed point using the polygonal approximation to the contour. These values are normalized to lie in $[0, 2\pi]$ and then stored in the vector

$$\mathbf{t}_n = (t_{n1}, t_{n2}, \ldots, t_{nJ})^T. \qquad (3)$$

Combining eqs.(1),(2) and (3), we can see that the point set $\mathbf{X}_n$ is approximated by evaluating the composite function $f_n(g_n(t))$ at the entries of $\mathbf{t}_n$. The same approximation can be written in terms of the long vectors $\mathbf{x}_n$ and $\mathbf{v}_n$ and expressed as a generalized linear regression on $g_n(t)$:

$$\mathbf{x}_n \approx \boldsymbol{\Phi}_n \mathbf{v}_n, \qquad (4)$$

where $\boldsymbol{\Phi}_n \in \mathbb{R}^{2J \times 2K}$ is the design matrix defined as

$$\boldsymbol{\Phi}_n \equiv \begin{pmatrix} \boldsymbol{\Omega}_n & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega}_n \end{pmatrix}; \qquad [\boldsymbol{\Omega}_n]_{jk} \equiv \phi_k(g_n(t_{nj})). \quad (5)$$

By initializing each g_n to the identity function, we initially estimate the correspondences using arc length. During learning, the $\mathbf{t}_n$ remain fixed but the g_n change. We are now in a position to define the probabilistic model.

3. Probabilistic Contour Model (PCM)

We assume that the intrinsic dimensionality of the data is low and accordingly, use a latent variable model with a low-dimensional latent space – Fig. 2e. Letting $\mathbf{z}_n \in \mathbb{R}^L$ be the latent variable associated with shape n (typically $L \ll K \ll J$), the prior distribution of the i.i.d. $\mathbf{z}_n$ is assumed to be a spherical Gaussian

$$\mathbf{z}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_L). \qquad (11)$$

The key component of the model is the conditional distribution

$$\mathbf{x}_n | \mathbf{z}_n \sim \mathcal{N}(\boldsymbol{\Phi}_n(\mathbf{W}\mathbf{z}_n + \boldsymbol{\mu}), \sigma^2 \mathbf{I}_{2J}), \qquad (12)$$

where $\sigma^2 \in \mathbb{R}$ is the noise variance, $\mathbf{W} \in \mathbb{R}^{2K \times L}$ is a linear mapping from latent space to coefficient space (which contains the $\mathbf{v}_n$ introduced in the previous section) and $\boldsymbol{\mu} \in \mathbb{R}^{2K}$ is the mean coefficient vector. There is independent Gaussian noise of equal magnitude on all output dimensions which amounts to a circular Gaussian about each of the J observed 2D points. It is easily shown that the marginal distribution of $\mathbf{x}_n$ is

$$\mathbf{x}_n \sim \mathcal{N}(\boldsymbol{\Phi}_n \boldsymbol{\mu}, \boldsymbol{\Phi}_n \mathbf{W}\mathbf{W}^T \boldsymbol{\Phi}_n^T + \sigma^2 \mathbf{I}_{2J}), \qquad (13)$$

and that the posterior distribution is given by

$$\mathbf{z}_n | \mathbf{x}_n \sim \mathcal{N}(\mathbf{M}_n^{-1} \mathbf{W}^T \boldsymbol{\Phi}_n^T (\mathbf{x}_n - \boldsymbol{\Phi}_n \boldsymbol{\mu}), \sigma^2 \mathbf{M}_n^{-1}), \quad (14)$$

where

$$\mathbf{M}_n \equiv \mathbf{W}^T \boldsymbol{\Phi}_n^T \boldsymbol{\Phi}_n \mathbf{W} + \sigma^2 \mathbf{I}_L. \qquad (15)$$

The generative model is strongly related to probabilistic principal component analysis (PPCA) (Bishop, 2006). However, rather than learning a linear mapping from the latent space directly to the data space, we map the latent space to the coefficients of the curve. The curve associated with these coefficients is then evaluated at the warped time points $g_n(t_{n1}), g_n(t_{n2}), \ldots, g_n(t_{nJ})$ and the data points are generated by small isotropic Gaussians centered at each of the evaluated points. Note that once the parameters have been estimated, all new curves generated from the model are functions of the same 'reference time' parameter (*i.e.* they are correctly corresponded) and need not be reparameterized.

Table 1. ECM algorithm for learning a Probabilistic Contour Model (PCM).

E-step Compute the sufficient statistics using the current parameter values:

$$\mathbb{E}[\mathbf{z}_n] = \mathbf{M}_n^{-1}\mathbf{W}^T\mathbf{\Phi}_n^T(\mathbf{x}_n - \mathbf{\Phi}_n\boldsymbol{\mu}), \qquad \mathbb{E}[\mathbf{z}_n\mathbf{z}_n^T] = \sigma^2\mathbf{M}_n^{-1} + \mathbb{E}[\mathbf{z}_n]\mathbb{E}[\mathbf{z}_n]^T \tag{6}$$

CM-steps Update the parameters using the sufficient statistics; repeat E-step between each CM step. Numerical optimization (Levenberg-Marquardt algorithm) is used in eq.(10). The $\otimes$ symbol denotes the Kronecker product.

$$\sigma^2_{new} = \frac{1}{2NJ}\sum_{n=1}^{N}\left(\|\mathbf{x}_n - \mathbf{\Phi}_n\boldsymbol{\mu}\|^2 - 2\mathbb{E}[\mathbf{z}_n]^T\mathbf{W}^T\mathbf{\Phi}_n^T(\mathbf{x}_n - \mathbf{\Phi}_n\boldsymbol{\mu}) + \mathrm{Tr}(\mathbb{E}[\mathbf{z}_n\mathbf{z}_n^T]\mathbf{W}^T\mathbf{\Phi}_n^T\mathbf{\Phi}_n\mathbf{W})\right) \tag{7}$$

$$\boldsymbol{\mu}_{new} = \left(\sum_{n=1}^{N}\mathbf{\Phi}_n^T\mathbf{\Phi}_n\right)^{-1}\sum_{n=1}^{N}\mathbf{\Phi}_n^T(\mathbf{x}_n - \mathbf{\Phi}_n\mathbf{W}\mathbb{E}[\mathbf{z}_n]) \tag{8}$$

$$\mathrm{vec}(\mathbf{W}_{new}) = \left(\sum_{n=1}^{N}\mathbb{E}[\mathbf{z}_n\mathbf{z}_n^T]\otimes\mathbf{\Phi}_n^T\mathbf{\Phi}_n\right)^{-1}\mathrm{vec}\left(\sum_{n=1}^{N}\mathbf{\Phi}_n^T(\mathbf{x}_n - \mathbf{\Phi}_n\boldsymbol{\mu})\mathbb{E}[\mathbf{z}_n]^T\right) \tag{9}$$

$$\{\boldsymbol{\alpha}_n, c_n\}_{new} = \arg\min_{\{\boldsymbol{\alpha}_n, c_n\}} -\|\mathbf{x}_n - \mathbf{\Phi}_n\boldsymbol{\mu}\|^2 + 2\mathbb{E}[\mathbf{z}_n]^T\mathbf{W}^T\mathbf{\Phi}_n^T(\mathbf{x}_n - \mathbf{\Phi}_n\boldsymbol{\mu}) - \mathrm{Tr}(\mathbb{E}[\mathbf{z}_n\mathbf{z}_n^T]\mathbf{W}^T\mathbf{\Phi}_n^T\mathbf{\Phi}_n\mathbf{W}) \tag{10}$$

Procrustes Step Rotate and translate each point set $\mathbf{X}_n$ so as to minimize $\|\mathbf{x}_n - \mathbf{\Phi}_n(\mathbf{W}\mathbb{E}[\mathbf{z}_n] - \boldsymbol{\mu})\|^2$ (recall that $\mathbf{x}_n \equiv \mathrm{vec}(\mathbf{X}_n)$). This is a Procrustes problem with a simple closed form solution (*e.g.* Cootes and Taylor (1999)).

Having considered all the distributions, the next step is to estimate the model parameters. In PPCA, the maximum likelihood estimates (MLEs) of the parameters can be computed in closed form or using the EM algorithm (Bishop, 2006). Here, the presence of the RFs complicates parameter estimation and there is no closed form solution for the MLEs and no exact M-step for the EM algorithm. However, minimizing the expectation of the complete log-likelihood with respect to one of $\mathbf{W}, \boldsymbol{\mu}, \sigma^2, \{\boldsymbol{\alpha}_1, c_1, \ldots, \boldsymbol{\alpha}_N, c_N\}$ while holding the others fixed is reasonably straight forward and leads to the Expectation Conditional Maximization (ECM) algorithm (Bishop, 2006) in Table 1. Note that there is no closed form expression for $\{\boldsymbol{\alpha}_n, c_n\}_{new}$ (eq.(10)) so a nonlinear optimization technique is required; in all experiments, the Levenberg-Marquardt algorithm was used for this minimization and the trapezoidal rule was used to evaluate the RFs (eq.(2)).

As with other approaches, it is assumed that scale, translation and rotation are nuisance transformations that do not alter the actual shape of a contour. Rather than complicating the model by including these transformations in the conditional distribution $p(\mathbf{x}_n|\mathbf{z}_n)$ (eq.(12)), we have found that transforming each point set $\mathbf{X}_n$ so as to maximize the expected complete log-likelihood works well in practice, *i.e.* we maximize the same objective function used for parameter estimation, but transform the data shapes rather than the model. Since the posterior distribution $p(\mathbf{z}_n|\mathbf{x}_n)$ is Gaussian, this maximization reduces to a standard Procrustes matching problem between $\mathbf{X}_n$ and the point set corresponding to the *maximum a posteriori* (MAP) estimate of $\mathbf{z}_n$. Scale is not included in the optimization since this would enable the model to produce a high likelihood by simply shrinking all the point sets. To remove the impact of scale, all point sets are normalized to have equal size, where size is defined as the mean squared distance from the centroid.

4. Nonlinear PCM (NPCM)

In PCM, the marginal distribution $p(\mathbf{x}_n)$ is Gaussian (eq.(13)) and hence, the distribution of each 2D boundary point is Gaussian. Such a model is not suitable for data sets displaying non-linear variation (*e.g.* where a single point traces out a curved path), but this limitation is not unique to PCM; to the best of our knowledge, none of the existing algorithms for *learning the correspondences and the model simultaneously* are designed to handle nonlinear shape variation. In contrast, a variety of techniques have been proposed for handling complex shape variation *when the correspondence between training shapes is known.* For example, one can fit a mixture of Gaussians to a low dimensional representation of the data rather than a single Gaussian (Cootes & Taylor, 1999). Alternatively, one can focus on the mapping between data space and latent space. This is the approach taken by Twining and Taylor (2001), where standard PCA is replaced with kernel PCA.

In this section, we extend PCM to nonlinear PCM (NPCM) by allowing the mapping from latent space to coefficient space to be nonlinear. This is achieved by

Table 2. ECM algorithm for learning a Nonlinear Probabilistic Contour Model (NPCM).

E-step Compute the responsibilities using the current parameter values and eq.(20): $p(\mathbf{z}_g|\mathbf{x}_n) = \frac{p(\mathbf{x}_n|\mathbf{z}_g)}{\sum_g p(\mathbf{x}_n|\mathbf{z}_g)}$.

CM-steps Update the parameters; numerical optimization (Levenberg-Marquardt algorithm) is used in eq.(19).

$$\sigma^2_{new} = \frac{1}{2NJ}\sum_{n,g} p(\mathbf{z}_g|\mathbf{x}_n)\|\mathbf{x}_n - \mathbf{\Phi}_n(W\boldsymbol{\gamma}(\mathbf{z}_g)+\boldsymbol{\mu})\|^2 \tag{16}$$

$$\boldsymbol{\mu}_{new} = \left(\sum_{n=1}^{N}\left(\sum_{g=1}^{G} p(\mathbf{z}_g|\mathbf{x}_n)\right)\mathbf{\Phi}_n^T\mathbf{\Phi}_n\right)^{-1}\sum_{n,g} p(\mathbf{z}_g|\mathbf{x}_n)\mathbf{\Phi}_n^T(\mathbf{x}_n - \mathbf{\Phi}_n W\boldsymbol{\gamma}(\mathbf{z}_g)) \tag{17}$$

$$\text{vec}(\mathbf{W}_{new}) = \left(\sum_{n,g} p(\mathbf{z}_g|\mathbf{x}_n)\left(\boldsymbol{\gamma}(\mathbf{z}_g)\boldsymbol{\gamma}^T(\mathbf{z}_g)\right)\otimes\left(\mathbf{\Phi}_n^T\mathbf{\Phi}_n\right) + \frac{\sigma^2}{\beta^2}\mathbf{I}_{2KU}\right)^{-1}\text{vec}\left(\sum_{n,g} p(\mathbf{z}_g|\mathbf{x}_n)\mathbf{\Phi}_n^T(\mathbf{x}_n - \mathbf{\Phi}_n\boldsymbol{\mu})\boldsymbol{\gamma}^T(\mathbf{z}_g)\right) \tag{18}$$

$$\{\boldsymbol{\alpha}_n, c_n\}_{new} = \arg\min_{\{\boldsymbol{\alpha}_n,c_n\}}\sum_g p(\mathbf{z}_g|\mathbf{x}_n)\|\mathbf{x}_n - \mathbf{\Phi}_n(W\boldsymbol{\gamma}(\mathbf{z}_g)+\boldsymbol{\mu})\|^2 \tag{19}$$

Procrustes Step Rotate and translate each $\mathbf{X}_n$ to match the point set associated with $\arg\max_{\mathbf{z}_g} p(\mathbf{z}_g|\mathbf{x}_n)$.

replacing the conditional distribution in eq.(12) with

$$\mathbf{x}_n|\mathbf{z}_n \sim \mathcal{N}(\mathbf{\Phi}_n(\mathbf{W}\boldsymbol{\gamma}(\mathbf{z}_n)+\boldsymbol{\mu}), \sigma^2\mathbf{I}_{2J}), \tag{20}$$

where

$$\boldsymbol{\gamma}(\mathbf{z}_n) \equiv (\gamma_1(\mathbf{z}_n), \gamma_2(\mathbf{z}_n), \ldots, \gamma_U(\mathbf{z}_n))^T \tag{21}$$

is a vector of basis functions with $\gamma_u(\mathbf{z})$:$\mathbb{R}^L \to \mathbb{R}$ and $\mathbf{W}$ is now a $2K \times U$ matrix. Just as PCM is related to PPCA (Sec. 3), NPCM is related to a well-known nonlinear extension of PPCA: the Generative Topographic Mapping (GTM) (Bishop, 2006).

The nonlinearity introduced in eq.(20) allows the marginal distribution $p(\mathbf{x}_n)$ to be non-Gaussian, enabling us to capture more complex types of shape variation. However, eq.(20) also complicates the expression for the log-likelihood and we can no longer derive a clean ECM algorithm (*cf.* Table 1). As in GTM, we overcome this by switching from a Gaussian prior to a discretized uniform prior:

$$p(\mathbf{z}_n) = \frac{1}{G}\sum_{g=1}^{G}\delta(\mathbf{z}_n - \mathbf{z}_g), \tag{22}$$

where the $\mathbf{z}_g$ are grid points of the latent space. For each n, the grid prior gives rise to a constrained G-component Gaussian mixture model in data space: each $\mathbf{z}_g$ is mapped to a coefficient vector $\mathbf{W}\boldsymbol{\gamma}(\mathbf{z}_g)+\boldsymbol{\mu}$, the curve associated with these coefficients is evaluated at $g_n(t_{nj})$ $(j = 1, 2, \ldots, J)$ and a spherical Gaussian is placed at each of the J 2D points. Since $\mathbf{x}_n \in \mathbb{R}^{2J}$, each component of the mixture model is formally a spherical Gaussian of dimension $2J$.

As discussed above, the linear PCM model may be *too simple* and unable to explain shape variation *given the correct RFs.* Conversely, the danger with NPCM is that the model may be *too complex* and will be able explain shape variation *given incorrect RFs.* The natural solution to this problem is to guide the algorithm towards smooth mappings by regularizing $\mathbf{W}$. Again, we follow the approach used in GTM and define a radially-symmetric Gaussian prior over the entries of $\mathbf{W}$:

$$p(\mathbf{W}|\beta^2) = \left(\frac{1}{2\pi\beta^2}\right)^{UK}\exp\left\{-\frac{1}{2\beta^2}\sum_{k=1}^{2K}\sum_{u=1}^{U} w_{ku}^2\right\}. \tag{23}$$

The matrix $\mathbf{W}$ is now a latent variable rather than a point parameter (the graphical model for NPCM is otherwise unchanged from that for PCM – Fig. 2e). To avoid integrating over $\mathbf{W}$ we use its MAP estimate (eq.(18), Table 2). The model parameters are estimated using the ECM algorithm in Table 2.[1]

5. Implementation

To implement PCM, the user must specify the following parameters:

L= # latent space dimensions (1; except in Sec. 6.2).
K= # curve basis functions for x and y coords. (50).
Q = # RF basis functions (8).

The numbers in brackets are the values used in our experiments. The low-dimension of the latent space combined with the noise model essentially regularizes the curves – it decides what constitutes genuine shape variation. Thus, we can choose a large K and not

[1] A hard assignment is used during the Procrustes step to improve efficiency.

worry about overfitting. The RFs are not regularized but this could easily be incorporated. The choice of L is an important model selection problem and future work will consider automatic selection of this parameter.

In all experiments, von Mises shaped unimodal periodic basis functions were used for both the curves and the RFs: $\phi(t;\rho_q,\kappa) = \mathrm{e}^{\kappa\cos(t-\rho_q)} / \mathrm{e}^{\kappa}$, where ρ_q is the center and κ is a width parameter. Given K (or Q), the centers were placed at equal intervals in $[0, 2\pi]$ and κ was fixed at $K/2$ (or $Q/2$). Note that multiscale RFs could be investigated by including basis functions of different widths.

To initialize PCM, the point sets are aligned to a common mean (with respect to the initial correspondences) using generalized Procrustes matching, a 2D curve is fitted to each point set and then PPCA on the coefficient vectors is used to initialize $\boldsymbol{\mu}$, σ and $\mathbf{W}$.[2] The RFs are initialized to the identity function by setting $\boldsymbol{\alpha}_n$=$\mathbf{0}$ and c_n=0 (eq.(2)). The data sets used in Sec. 6 were chosen to enable comparison with other techniques and in all cases, the first point of each shape is correctly corresponded across training examples. Since this information may not be available in some applications, it is worth noting that PCM/NPCM only requires rough initial correspondences and that there are accurate, efficient algorithms for finding these (*e.g.* (McNeill & Vijayakumar, 2006)).

PCM is surprisingly robust and if a data set contains nonlinear variation, it tends to produce as good a linear model as we might hope for rather than failing completely. This suggests using PCM to initialize NPCM which can be achieved by setting the first L entries of $\boldsymbol{\gamma}(\mathbf{z})$ in eq.(21) equal to the entries of $\mathbf{z}$, setting the first L columns of $\mathbf{W}$ in eq.(20) equal to the appropriately scaled $\mathbf{W}$ from PCM (eq.(12)) and the remaining entries to zero. There are alternative ways in which PCM could be used to initialize NPCM but we have found this approach to work well in practice. It is straight forward to assign a different prior variance to the entries of $\mathbf{W}$ associated with different types of basis function, but in our experiments we used the single default value given below. The additional parameters required for NPCM are:

G = # grid points in lat. space (50; except in Sec. 6.2).
U = # basis functions for lat. space → coeff. space mapping (1 linear + 8 radial basis functions (RBFs); except in Sec. 6.2).
β = prior variance on the entries of $\mathbf{W}$ (0.1).

Ideally, G should be large in order to accurately approximate a continuous latent space, but this must be balanced against the associated rise in computation time. Gaussian shaped RBFs were used with centers placed on a uniform grid containing $U - L$ vertices and with the variance fixed so that neighboring centers were two standard deviations apart.

The algorithms were implemented in Matlab on a 1.6GHz Intel Centrino Duo machine. In all experiments, 200 EM iterations were used for both PCM and NPCM (200 E-steps, 40 each of the CM/Procrustes steps). Learning the shape model for a data set of 22 shapes, each described by 128 points took <3min for PCM and <9min for NPCM.

6. Experiments and Evaluations

6.1. Illustrative Examples: Box-bump Data

Fig. 1a shows a ‘box-bump’ data set similar to those used by Davies (2002), Hladuvka and Buhler (2005) and Ericsson and Karlsson (2006). Note that unlike most natural data sets, the distribution of the shapes is uniform given the correct correspondence (and ignoring minor effects due to alignment transformations). For example, the point at the top of the bump is located at equal increments along a straight line as the bump moves from left to right. The same is true of all points on the top of the shape (including non-bump points), whereas points on the side and base of the shapes do move at all. Given this uniform distribution in data space and the fact that PCM is a linear model, we would expect the posterior means of the training shapes to be uniformly distributed in latent space if the model is accurate. Fig. 1 shows the results of learning a model with fixed RFs and demonstrates, as has been done in the past, that arc length parameterization produces a poor shape model for this type of data. Note that the mean shape (Fig. 1b, center) has a low, elongated bump, and the posterior means are neither uniformly distributed nor do they reflect the Gaussian prior (Fig. 1c).

The results of applying PCM to the box-bumps are summarized in Fig. 2. The slight rotations of the shapes in the figures are those applied by PCM during the Procrustes step; as would be expected, the generated shapes are similarly rotated. These alignment transformations are included in the figures for completeness, but note that the orientation of the shapes is of no importance when assessing the accuracy of the shape model. The learnt correspondences in Fig. 2a are better than those associated with arc length in

[2] For many choices of basis function, isotropic noise on the coefficients does not equate to noise of equal magnitude at each boundary point. This is one of numerous problems associated with modeling curve coefficients directly.

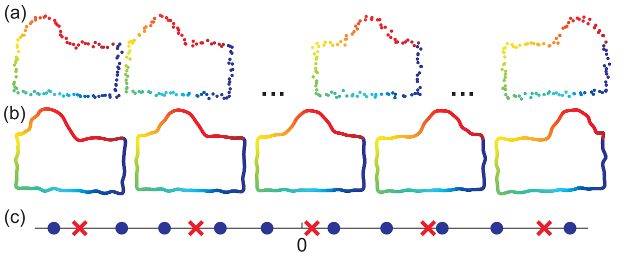

Figure 3. (a) Noisy box-bump data with learnt parameterizations. (b) Curves associated with equal increments in latent space. (c) Latent space.

Fig. 1a – the color distribution on each bump is approximately the same. The shape of the RFs in Fig. 2d indicates that PCM has captured the point wise variation discussed in the previous paragraph. Fig. 2c (circles) shows the 1D latent space with each training shape represented by its posterior mean – *i.e.* the mean of the Gaussian in eq.(14). Note that the circles are uniformly distributed in latent space despite the Gaussian prior. Rather than projecting from data space to latent space, we can go in the opposite direction by selecting a point in latent space and generating the corresponding curve. The curves in Fig. 2b correspond to the uniformly spaced crosses in Fig. 2c. Note that these resemble the training shapes (Fig. 2a), unlike those generated from the arc length model (Fig. 1b).

In Fig. 3a, each point of the box-bump shapes has been contaminated with independent Gaussian noise of standard deviation 0.3. PCM is robust to this type of noise by construction and learns a reasonably accurate model with an estimated standard deviation of 0.26. Note that the generated shapes in Fig. 3b reflect the true shape variation despite the large noise variance and there being only 10 training shapes.

6.2. Dimensionality Reduction/Visualization

In Fig. 4, PCM/NPCM is used to visualize a data set of 23 shapes – the 12 shark shapes used in Sec. 6.3 and the 11 fish/shark shapes from Kimia's data set (Sebastian et al., 2004). For the 2D NPCM model we used G=25^2 and U=27 (2 linear basis functions and 5^2 RBFs). The standard 200 iterations were used but it is worth noting that the posterior distributions required for visualization typically converge after much fewer iterations. The distribution of posterior means in Figs. 4b and 4c is perhaps more intuitive than in 4a, but more importantly, 4b and 4c are based on variation between homologous morphological features rather than artificial variation arising from a naive arc length parameterization – see Sec. 6.3 and the second column of Table 4 in particular.

6.3. Benchmark Data Sets

Ericsson and Karlsson (2006) recently evaluated state-of-the-art algorithms using a "ground truth correspondence measure" (GCM) which avoids problems associated with the compactness, specificity and generality measures used by Davies (2002). To compute the GCM, multiple independent observers identify a prespecified set of landmarks on each shape. The resulting distribution of landmarks is then compared to the estimated landmark positions given by the algorithm under evaluation. In simple terms, the GCM is the error in an algorithm's approximation of the ground truth. Table 3 shows the GCM values for five shape classes. The second row gives the GCM for an arc length parameterization and subsequent rows give the GCM for different algorithms *as a percentage of the arc length GCM*. The algorithms tested include variants of the MDL approach: the "cur" algorithms use curvature, algorithms 3-5 use different techniques to prevent degenerate solutions (point clustering – a problem that does not arise with PCM/NPCM) and the AIAS+MDL algorithms combine MDL with an affine invariant approach (Ericsson & Astrom, 2003a). Note that different algorithms perform well on different data sets. For example, the AIAS+MDL algorithms perform very well on the birds data and curvature information seems to be useful for the f.birds data. This suggests that low-level choices regarding transformation invariance and shape features are important but *application dependent.*

The 128-point shapes and ground truth information used by Ericsson and Karlsson (2006) was used to compute the results in Table 4.[3] Aside from the poor performance on the rats data, PCM performs reasonably well and NPCM performs very well. It is important to note at this point that PCM/NPCM is *not* a variant of an existing technique (*c.f.* Table 3). Rather, it is a novel approach to learning shape models with *application independent* advantages over existing techniques (Sec. 1). As with MDL, variants of the basic PCM/NPCM algorithms (*e.g.* incorporating curvature) could easily be introduced.

[3] Six closed curve data sets were provided but we had difficulty processing the box-bump data and ground truth, so this data set is omitted (but see Sec. 6.1). Also, we are trying to identify the cause of small discrepancies between the arc length GCM values reported here and those published by Ericsson and Karlsson (2006) (0.1-3.4% – 2nd row, Tables 3 and 4).

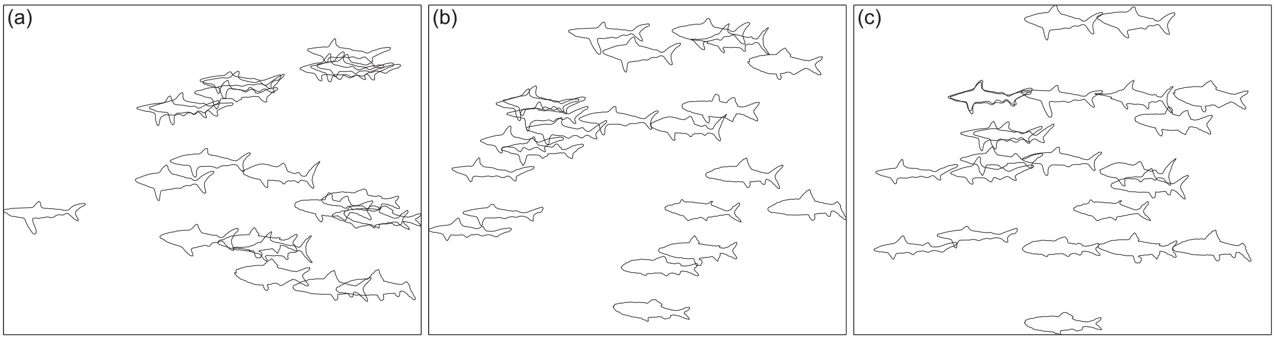

Figure 4. Each data shape is shown at the position of its posterior mean. (a) PCM with no RFs. (b) PCM. (c) NPCM.

Table 3. Ground Truth Correspondence Measure for existing algorithms – from Ericsson and Karlsson (2006).

Algorithm	sharks	birds	f.birds	rats	forks
Arc length	15.55	22.88	7.12	13.37	19.11
Resid. error (%):					
1.MDL	27	65	56	29	19
2.MDL,cur	22	80	45	27	23
3.MDL,me	29	92	62	30	20
4.MDL,nodecost	26	67	56	29	19
5.MDL,par	24	62	48	28	18
6.AIAS,MDL	22	23	58	28	20
7.AIAS,MDL,cur	22	24	48	27	24
8.Eucl	44	60	59	37	27
9.Eucl,cur	29	55	54	35	29
10.Cur	22	111	46	29	31

Table 4. Ground Truth Corresp. Measure: PCM/NPCM.

Algorithm	sharks	birds	f.birds	rats	forks
Arc length	15.65	22.91	6.93	13.84	19.19
Resid. error (%):					
1.PCM	27	56	52	76	20
2.NPCM	21	42	47	51	15

7. Summary and Future Work

We have presented a generative probabilistic approach to modeling shape contours which overcomes many of the problems associated with existing algorithms. Future work will investigate mechanisms for handling outliers in both latent space (outlying shapes), and data space (outlying points, often due to occlusion or missing parts). One possibility is to assume that each data point is generated by either the PCM/NPCM model or a high variance outlier distribution. This mixture model approach to handling outliers has been successfully applied to *unordered* point sets in the past (*e.g.* Chui and Rangarajan (2000)). Other possibilities for future work include automated model selection and extensions to 3D data.

Acknowledgements

We thank Anders Ericsson and Johan Karlsson for supplying the data and ground truth used in Sec. 6.3.

References

Bishop, C. (2006). *Pattern recognition and machine learning*. Springer.

Chui, H., & Rangarajan, A. (2000). A feature registration framework using mixture models. *MMBIA*.

Cootes, T., & Taylor, C. (1999). A mixture model for representing shape. *Im. and Vis. Comp.*, *17*, 567–574.

Davies, R. (2002). *Learning shape: Optimal models for analysing shape variability*. Doctoral dissertation, University of Manchester.

Davies, R. H., Twining, C. J., Cootes, T. F., Waterton, J. C., & Taylor, C. J. (2002). A minimum description length approach to statistical shape modeling. *IEEE Trans. on Med. Im.*, *21*, 525–537.

Ericsson, A., & Astrom, K. (2003a). An affine invariant deformable shape representation for general curves. *ICCV*.

Ericsson, A., & Astrom, K. (2003b). Minimizing the description length using steepest descent. *BMVC*.

Ericsson, A., & Karlsson, J. (2006). Benchmarking of algorithms for automatic correspondence of shapes. *BMVC*.

Hladuvka, J., & Buhler, K. (2005). MDL spline models: Gradient and polynomial reparameterisations. *BMVC*.

Kotcheff, A. C. W., & Taylor, C. J. (1998). Automatic constuction of eigenshape models by direct optimization. *Med. Im. Anal.*, *2*, 303–314.

McNeill, G., & Vijayakumar, S. (2006). Hierarchical Procrustes matching for shape retrival. *CVPR*.

Ramsay, J., & Silverman, B. (2005). *Functional data analysis*. Springer.

Sebastian, T. B., Klein, P. N., & Kimia, B. B. (2004). Recognition of shapes by editing their shock graphs. *PAMI*, *26*, 550–571.

Thodberg, H. (2003). Minimum description length shape and appearance models. *IPMI*.

Thodberg, H., & Olafsdottir, H. (2003). Adding curvature to minimum description length shape models. *BMVC*.

Twining, C., & Taylor, C. (2001). Kernel principal component analysis and the construction of non-linear active shape models. *BMVC*.