Kernel Principal Component Analysis and the Construction of Non-Linear Active Shape Models

C. J. Twining and C. J. Taylor
Imaging Science and Biomedical Engineering,
The University of Manchester, Oxford Road, Manchester M13 9PT.
Email: **carole.twining@man.ac.uk**   URL: **http://www.isbe.man.ac.uk**

Abstract

The use of Kernel Principal Component Analysis (KPCA) to model data distributions in high-dimensional spaces is described. Of the many potential applications, we focus on the problem of modelling the variability in a class of shapes. We show that a previous approach to representing non-linear shape constraints using KPCA is not generally valid, and introduce a new 'proximity to data' measure that behaves correctly. This measure is applied to the building of models of both synthetic and real shapes of nematode worms. It is shown that using such a model to impose shape constraints during Active Shape Model (ASM) search gives improved segmentations of worm images than those obtained using linear shape constraints.

1   Introduction

Many computer vision problems involve modelling the distribution of data in high dimensional spaces. Our particular interest is in statistical shape modelling [3, 4], where the 'legal' variability in a class of shapes may be learnt by modelling the distribution of shape vectors over a training set. In many situations one can assume that the distribution can be modelled as a multivariate Gaussian, the parameters of which are obtained using linear principal component analysis [8]. In other cases, it is necessary to use a non-linear model. Sozou et al, Cootes et al and Heap et al [19, 20, 5, 7] showed previously that this is the case for some types of shape variability (e.g. large-amplitude bending), and proposed non-linear methods of modelling shape distributions. None of these approaches is, however, both general and robust.

Kernel principal component analysis (KPCA) [16, 17] is a technique for non-linear feature extraction, closely related to methods applied in Support Vector Machines [13]. It has proved useful for various applications, such as de-noising [9] and as a pre-processing step in regression problems [12]. KPCA has also been applied, by Romdhani et. al. [11], to the construction of non-linear statistical shape models of faces, but we will argue that their approach to constraining shape variability is not generally valid.

We propose a new method of constructing a 'proximity to data' measure, based on KPCA, and show that it does not suffer from the problems inherent in Romdhani's approach. The method is applied to both synthetic and real shape data and is shown to behave as predicted. The model derived from real shapes (images of nematode worms) is

used in Active Shape Model (ASM) search [3, 4] to impose non-linear shape constraints whilst iteratively deforming the shape template to improve the match between model and image features. The resulting segmentations are shown to be, on average, more accurate than those obtained when shape is constrained using a linear model. In the remainder of this paper we give a brief explanation of KPCA with Gaussian kernel functions, a discussion of the way in which training data are distributed in KPCA space and arguments to support our contention that the approach to modelling shape constraints proposed by Romdhani et. al. [11] is not generally valid. We introduce a new 'proximity to data' measure, and present experimental results for both synthetic and real data.

2 Kernel Principal Component Analysis

Kernel principal component analysis is a method of non-linear feature extraction. The non-linearity is introduced via a mapping of the data from the input space $\mathcal{I}$ to a feature space $\mathcal{F}$. Linear principal component analysis is then performed in the feature space; this can be expressed solely in terms of dot products in the feature space. Hence, the non-linear mapping need not be explicitly constructed, but can be specified by defining the form of the dot products in terms of a Mercer Kernel function $\mathcal{K}$ on $\mathcal{I} \times \mathcal{I}$. We concentrate on the case of a Gaussian kernel function.

2.1 The Basic KPCA Algorithm

Consider data points $\vec{x}$ and $\vec{y}$ [1] in the input space $\mathcal{I} = \mathbb{R}^n$. The non-linear mapping $\mathbf{\Phi} : \mathbb{R}^n \to \mathcal{F}$ is defined such that:

$$\mathbf{\Phi}\left(\vec{x}\right) \bullet \mathbf{\Phi}\left(\vec{y}\right) \equiv \mathcal{K}\left(\vec{x}, \vec{y}\right) = \exp\left(-\frac{1}{2\sigma^2} \| \left(\vec{x} - \vec{y}\right) \|^2\right) \ \ \forall \ \vec{x}, \vec{y} \in \mathbb{R}^n \quad (1)$$

where $\bullet$ is the vector dot product in the infinite-dimensional feature space $\mathcal{F}$, and σ is the width of the kernel. Note that $\mathbf{\Phi}\left(\mathbb{R}^n\right)$ is an embedded sub-manifold of $\mathcal{F}$. Properties of this embedding [1] will become important later.

For a data set $\{\vec{x}_i : i = 1 \text{ to } N\}$, we have the corresponding set of mapped data points $\{\mathbf{\Phi}_i = \mathbf{\Phi}\left(\vec{x}_i\right) : i = 1 \text{ to } N\}$ in the feature space $\mathcal{F}$. Centered data points in $\mathcal{F}$ are defined thus:

$$\widetilde{\mathbf{\Phi}}\left(\vec{y}\right) \equiv \mathbf{\Phi}\left(\vec{y}\right) - \frac{1}{N}\sum_{i=1}^{N}\mathbf{\Phi}\left(\vec{x}_i\right) \ \ \forall \ \vec{y} \in \ \mathbb{R}^n. \quad (2)$$

Following from the definition of the dot product (1), the unnormalised and normalised Kernel matrices over the data set are defined thus:

$$\mathbf{K}_{ij} \ \equiv \ \mathbf{\Phi}\left(\vec{x}_i\right) \bullet \mathbf{\Phi}\left(\vec{x}_j\right) = \exp\left(-\frac{1}{2\sigma^2}\|\vec{x}_i - \vec{x}_j\|^2\right) \quad (3)$$

$$\widetilde{\mathbf{K}}_{ij} \ \equiv \ \widetilde{\mathbf{\Phi}}(\vec{x}_i) \bullet \widetilde{\mathbf{\Phi}}(\vec{x}_j). \quad (4)$$

$$= \ \mathbf{K}_{ij} - \frac{1}{N}\sum_{p=1}^{N}\mathbf{K}_{ip} - \frac{1}{N}\sum_{q=1}^{N}\mathbf{K}_{qj} + \frac{1}{N^2}\sum_{p,q=1}^{N}\mathbf{K}_{pq}.$$

---

[1]The notation $\vec{x}$ is used to denote vectors in a finite-dimensional space, whilst bold vectors (e.g. $\mathbf{\Phi}\left(\vec{x}\right)$) are used to distinguish those in infinite-dimensional spaces.

Linear principal component analysis is then performed by finding a set of vectors $\{\mathbf{b}^\alpha\}$ in the span of $\{\widetilde{\boldsymbol{\Phi}}\,(\vec{x}_i)\}$ which represent the principal axes of the data set, hence are the set of extrema of the Lagrangian:

$$\mathcal{L} = \frac{1}{2N} \sum_{i=1}^{N} \left[\mathbf{b}^\alpha \bullet \widetilde{\boldsymbol{\Phi}}\,(\vec{x}_i) \right]^2 - \frac{\lambda^\alpha}{2} \left[\mathbf{b}^\alpha \bullet \mathbf{b}^\alpha - a^\alpha \right] \tag{5}$$

where a^α determines the normalisation of the particular vector $\mathbf{b}^\alpha$, and the Lagrange multiplier $\lambda^\alpha > 0$. Setting the derivatives of $\mathcal{L}$ with respect to $\mathbf{b}^\alpha$ to zero, and taking the dot product with $\widetilde{\boldsymbol{\Phi}}\,(\vec{x}_j)$ gives the eigenvector equation:

$$\lambda^\alpha \mathbf{b}_j^\alpha = \frac{1}{N} \sum_i \widetilde{\mathbf{K}}_{ij} \mathbf{b}_i^\alpha \;\; \text{where} \;\; \mathbf{b}_k^\alpha \equiv \mathbf{b}^\alpha \bullet \widetilde{\boldsymbol{\Phi}}\,(\vec{x}_k). \tag{6}$$

In contrast to Mika et. al. [9] and Schölkopf et. al. [17], we find it convenient to normalise the eigenvectors with respect to the data thus:

$$\sum_{i=1}^{N} (\mathbf{b}_i^\alpha)^2 = 1 \;\; \forall \;\; \alpha. \tag{7}$$

It follows from the definition of $\widetilde{\mathbf{K}}_{ij}$ (5) that:

$$\sum_{j=1}^{N} \widetilde{\mathbf{K}}_{ij} \equiv 0, \;\; \text{hence} \;\; \sum_{j=1}^{N} \mathbf{b}_j^\alpha \equiv 0 \;\; \forall \;\; \alpha. \tag{8}$$

For the corresponding vectors $\mathbf{b}^\alpha$ in the feature space :

$$\mathbf{b}^\alpha = \frac{1}{N\lambda^\alpha} \sum_{i=1}^{N} \mathbf{b}_i^\alpha \widetilde{\boldsymbol{\Phi}}(\vec{x}_i) \equiv \frac{1}{N\lambda^\alpha} \sum_{i=1}^{N} \mathbf{b}_i^\alpha \boldsymbol{\Phi}(\vec{x}_i), \;\;\; \|\mathbf{b}^\alpha\|^2 = \frac{1}{N\lambda^\alpha}. \tag{9}$$

Hence the set of $\mathbf{b}^\alpha$ thus defined form an orthogonal but *not* an orthonormal basis for the space spanned by $\{\widetilde{\boldsymbol{\Phi}}\,(\vec{x}_i)\}$. Given a set of solutions $\{\mathbf{b}_i^\alpha : i = 1 \text{ to } N, \alpha = 1 \text{ to } M\}$, ordered in terms of non-increasing eigenvalue, we define an M-dimensional KPCA space as follows. A test point $\vec{x} \in \mathbb{R}^n$ is mapped to a point $\vec{p}\,(\vec{x})$ in this space, with unnormalised KPCA components :

$$\begin{aligned} p^\alpha\,(\vec{x}) \;\; &\equiv \;\; N\lambda^\alpha \mathbf{b}^\alpha \bullet \boldsymbol{\Phi}\,(\vec{x}) = \sum_i \mathbf{b}_i^\alpha \mathcal{K}\,(\vec{x}, \vec{x}_i) \\ &= \;\; \sum_i \mathbf{b}_i^\alpha \exp\left(-\frac{1}{2\sigma^2} \|\,(\vec{x} - \vec{x}_i)\,\|^2 \right) \end{aligned} \tag{10}$$

Note that normalised components can also be defined, by substituting $\widetilde{\boldsymbol{\Phi}}$ for $\boldsymbol{\Phi}$ in the above equation, which corresponds to a translation of the origin in KPCA space.

3 KPCA Space

KPCA space is the M-dimensional space of unnormalised non-linear components, where equation (10) gives us the values of the components of an arbitrary point $\vec{x}$ in the input

space. However, it should be noted that *not* all points in KPCA space correspond to points in input space. This is because, as noted previously, $\mathbf{\Phi}\left(\mathbb{R}^n\right)$ forms an embedded sub-manifold in the feature space $\mathcal{F}$. This can be easily demonstrated. The form of the mapping $\mathbf{\Phi}$ is determined by the definition of the Kernel function $\mathcal{K}$ (1). From this, it follows directly that:

$$\|\mathbf{\Phi}\left(\vec{x}\right)\|^2 \equiv \mathbf{\Phi}\left(\vec{x}\right) \bullet \mathbf{\Phi}\left(\vec{x}\right) = 1 \tag{11}$$

for all points $\vec{x}$. Hence $\mathbf{\Phi}\left(\mathbb{R}^n\right)$ must lie on some embedded sub-manifold in $\mathcal{F}$. However, it should be noted that this property of constancy of the modulus will *not* be preserved for points $\vec{p}\left(\vec{x}\right)$ when the sub-manifold is projected into KPCA space.

3.1 The Embedded Sub-manifold

Consider points in input space which are far from any data points. Directly from equation (10), it can be seen that *all* the non-linear components for these points will tend to zero as the points recede from the data. Hence, all points far from the data in input space will map to the vicinity of the origin in KPCA space.

Now consider points which have non-zero values of the components. The importance of the kernel width follows directly from equation (10); in the limit where σ is infinitesimal, the components will only be non-zero in the neighbourhood of some of the data points (those for which the $\mathbf{b}_i^\alpha$ are non-zero), where the size of this neighbourhood will be related to the kernel width. Also, the values of a component over the input space will take both negative and positive values, since, from equation (8) at least one of the $\mathbf{b}_i^\alpha$ must be of the opposite sign to the rest. These properties will persist up to some finite size of the kernel width. Hence, if we consider any single component $p^\alpha\left(\vec{x}\right)$, the extrema will tend to lie in the neighbourhood of some of the data points, and these extreme values will bracket the value obtained for points at infinity in input space. Since this is an ordering property, it will be preserved if we make the transition to normalised components, since this just corresponds to a translation of the origin. Furthermore, it follows from equations (9) and (11) that, as noted by Schölkopf et. al. [17], the unnormalised components $p^\alpha(\vec{x})$ are bounded thus:

$$|p^\alpha(\vec{x})| \leq N\lambda^\alpha \|\mathbf{b}^\alpha\| \|\mathbf{\Phi}\left(\vec{x}\right)\| = \sqrt{N\lambda^\alpha} \tag{12}$$

hence the embedded sub-manifold itself lies within a strictly bounded region of KPCA space. An example illustrating these points is given in Figure 1. It shows the sub-manifold and data points in KPCA space for a data set consisting of 100 equally spaced points on the unit circle in $\mathbb{R}^2$. A kernel width of 0.1 was used. Note that *all* the data points lie precisely at the periphery of the sub-manifold, and that the centre of the sub-manifold, corresponding to the origin of KPCA space, is bracketed by the data in all of the 6 dimensions shown.

So, we can now see that the property which distinguishes points in the vicinity of the data from all other points in input space, is that they lie near the periphery of the sub-manifold. Since the sub-manifold is bounded, and since points at the periphery bracket the origin whichever direction we consider, the distance from the origin in KPCA space can provide us with a 'proximity to data' measure. Consider the function $f\left(\vec{x}\right)$ on input space defined thus :

$$f\left(\vec{x}\right) = \sum_{\alpha=1}^{M} p^\alpha\left(\vec{x}\right) p^\alpha\left(\vec{x}\right) = \sum_{i,j,\alpha} \mathbf{b}_i^\alpha \mathbf{b}_j^\alpha \mathcal{K}\left(\vec{x},\vec{x}_i\right) \mathcal{K}\left(\vec{x},\vec{x}_j\right) \tag{13}$$
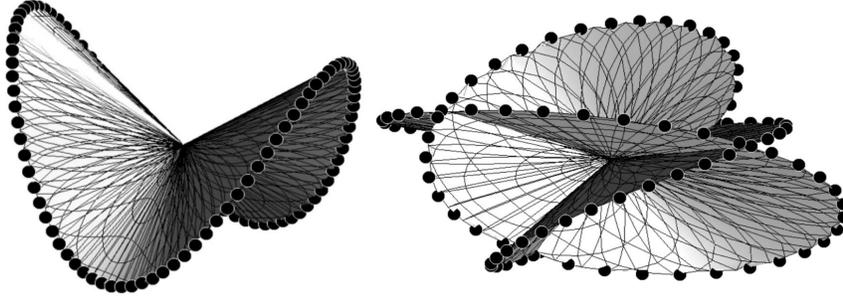
Figure 1: Plots of the embedded sub-manifold (surface) and data (filled circles) for the example data set. *Left*: First 3 KPCA components, *Right*: The 4th to 6th components.

which represents the square of the distance from the origin in KPCA space. This is a positive definite function, which, from equation (12), has a strict upper bound. The value of $f(\vec{x})$ will tend to zero as the point $\vec{x}$ recedes from the data, and the local maxima of $f(\vec{x})$ over input space will tend to lie in the neighbourhood of the data set.

4   Constructing Non-Linear ASMs

We now consider applying our 'proximity to data' measure $f(\vec{x})$ to training an Active Shape Model [3, 4]. An ASM requires a statistical shape model (SSM) that captures the allowed variability in a class of shapes. The ASM search algorithm requires that the SSM should allow the limits of valid variation to be defined, generalising from a training set to produce a continuous class of valid shapes, whilst excluding types of variation that are not present in the training set. A method is also required to calculate the nearest valid shape, given an invalid shape instance.

Each shape i in the training set is represented by a vector $\vec{x}_i$, constructed by concatenating the coordinates of a set of sample points around its boundary/boundaries. In the standard approach [6], principal component analysis is used to find the parameters of a multivariate Gaussian distribution that models the training data. Shape constraints can be applied by placing an *upper* bound on the modulus of each normalised principal component, that is, on the distance from the centre of the Gaussian (the mean shape) in the direction of the component, normalised by the associated standard deviation. The nearest valid instance to an input data point is constructed by finding the nearest point within the valid region which lies in the same direction as the input point.

In their construction of a non-linear ASM using KPCA, Romdhani et. al. [11] defined their valid shape regions by placing an *upper* bound on the modulus of each of the normalised KPCA components. In linear PCA the values of the components are zero when we are at the mean of the data, and the absolute values of the components increase without limit as we move away from the data. However, it can clearly be seen from section 3.1 that KPCA components *do not* behave in this manner; zero values of all components correspond to points far from the data, whilst the absolute values of all components are bounded, since the embedded sub-manifold is itself bounded.

We now apply the 'proximity to data' measure $f(\vec{x})$ defined above (13). A valid data region is defined by placing a *lower* bound on the allowed value of $f(\vec{x})$. The valid region

is then the interior of the specified isosurface. Given a point outside this region, it can be moved into the valid region by performing gradient ascent on $f(\vec{x})$. (Expressions for the derivatives of $f(\vec{x})$ can be calculated from equation (13)).

5 Experiments

5.1 Artificial Statistical Shape Model

We created a synthetic class of shapes, based on the nematode worm. Each shape consists of 16 segments (and associated landmark points), with the length and width of each worm segment fixed. Shapes are generated by specifying the angles between the segments. An ensemble of 200 training shapes was generated by setting the segment angles to be a linear function of segment number, with the addition of some Gaussian noise. Examples from the training set are shown in Figure 2. The values of the input parameters to the shape
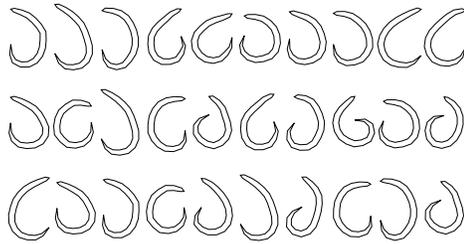
Figure 2: Examples from the training set of the artificial nematode SSM.
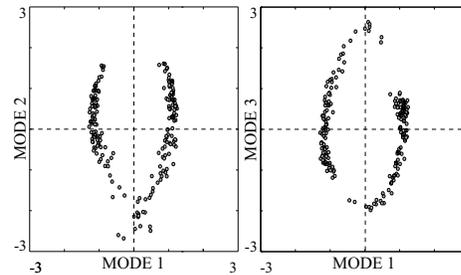
Figure 3: Scatter plots of the first 3 linear PCA components for the artificial nematode SSM, in units of standard deviations.

generator were chosen so that none of the training examples contained self-intersections. Linear PCA was then applied to the shape vectors obtained from the sets of landmark points of the training ensemble. It was found that 6 modes were enough to account for 99% of the variance across the training set. The distribution of the first 3 linear components is shown in Figure 3. It is clear that the distribution of the data is non-linear, and that, for example, the mean shape (the point with zero values for all modes), itself will not be a valid shape. Kernel PCA was then applied to the data points in the space of linear modes, and the 'proximity to data' measure constructed. A valid shape region was then defined, choosing a value of $f(\vec{x})$ such that all the training data lay within the defined isosurface. Gradient ascent to this level was then performed on a set of randomly chosen

Figure 4: Initial shapes (normal lines) and the recovered shapes (bold) for the artificial worm SSM.

28

Figure 5: Evolution from a random initial shape with self-intersections (left) to an aligned final shape (right) during gradient ascent.

shapes. A sample of the results are shown in Figure 4, and a typical evolution during gradient ascent in Figure 5. Note that the initial random shapes, as well as containing self-intersections, also show considerable variations in segment width. The procedure gives a reasonable resultant 'nearest' shape, smoothly removing the self-intersections and reducing the variations in width.

5.2 Non-Linear Active Shape Model

The training set of our ASM consisted of 30 images of the nematode worm *C. Elegans*, and their reflected images. Previous work on applying linear ASMs to these animals [2] had reported problems with segmenting the full range of worm images.

Each training image was labelled by hand with a set of 43 landmark points. An example labelled image is shown in Figure 6. A linear ASM was then built from this data. Thirteen linear modes were found to be sufficient to account for 99% of the shape variance. The profile models at each point were modelled linearly by a multivariate Gaussian. A non-linear ASM (but using the same linear profile models) was constructed by applying the 'proximity to data' measure to the space of linear shape components. This allows a direct comparison between linear and non-linear models to be made, since they are acting on the *same* dimensionally-reduced data. In Figure 7, we show the distribution of
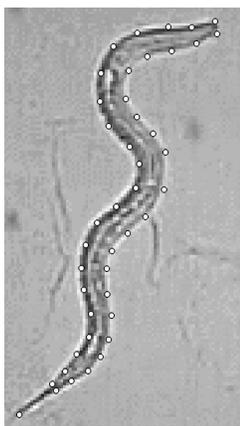
Figure 6: Close-up of a labelled image from the training set.
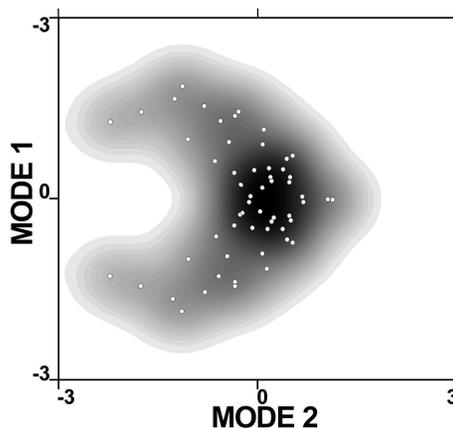
Figure 7: The nematode training set data across the first 2 linear modes (points), plus the exactly integrated 'proximity to data' measure (shading).

29

the training set across the first 2 linear modes, and also the 'proximity to data' measure, calculated with a kernel width of 0.5, exactly integrated over the remaining 11 modes. We then compared the linear and non-linear ASMs in search mode over the training set. Searching with an ASM is an iterative procedure. Given some configuration of the landmarks in the image, we then search in the neighbourhood of each point for a better fit to the profile model at that point. This set of suggested placements for the landmarks is then fitted as closely as possible by the model, whilst constraining the shape to lie within our valid shape region. This process is then iterated to convergence. It was found that
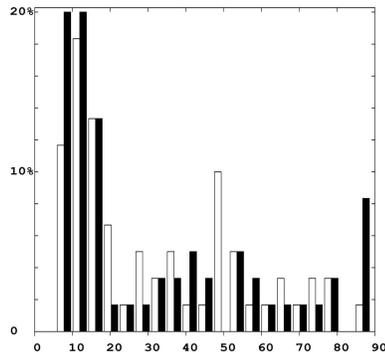
Figure 8: Distribution of the *square root* of the mean of the square of the point-to-point distance for linear (white bars) and non-linear (black bars) ASM searches.
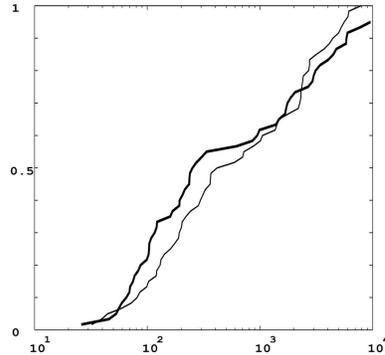
Figure 9: The cumulative frequency of the mean of the square point-to-point distance for linear (normal line) and non-linear (bold line) ASM searches.

better results were achieved if the non-linear shape constraints were only applied every few iterations, otherwise the linear constraints were applied. For each image, the starting configuration was taken as the mean shape from the linear ASM, aligned to the image labels. Once the search had converged, we then calculated the mean over the landmarks of the square point-to-point distance between the search results and the labels (See Figures 8 and 9).

Considering Figure 8, it can be seen that the leftmost peak of the distribution, corresponding to successful searches, is higher for the non-linear ASM. This result is also shown in Figure 9, where the initial portion of the cumulative frequency graph (0 to 0.6) for the non-linear ASM lies to the left of the corresponding graph for the linear ASM, corresponding to smaller resultant point-to-point distances. Note that the non-linear ASM tends to diverge for some failed searches (far right of figures). Hence we can conclude that the non-linear ASM offers an improvement in search performance over the standard ASM.

6   Summary and Conclusions

We have discussed the nature of Gaussian kernel PCA and it's application to non-linear data analysis. The problems with previous approaches to applying KPCA to the task of constructing non-linear ASMs [11] have been illustrated. We have shown how to construct a 'proximity to data' function in *input* space using KPCA components, which then avoids

the data reconstruction problem encountered when using KPCA components directly [9, 15]. We have demonstrated the applicability of this function using both artificial and real-world non-linear data examples. Finally, we have demonstrated an improvement in search performance when compared with a standard linear ASM.

What is still unresolved is how to optimise the kernel PCA parameters; the kernel width and the number of non-linear components retained in the calculation of $f(\vec{x})$. Also, KPCA, unlike corresponding linear techniques, does not give us an explicit parameterisation of the input data, and the topology of the isosurfaces of $f(\vec{x})$ is not obvious. We anticipate that the possible occurrence of disconnected regions will lead to problems with naive ASM search algorithms. It would obviously be desirable to place the choice of isosurface of $f(\vec{x})$ into a proper probabilistic framework. However, we anticipate this will present difficulties, since the form of $f(\vec{x})$ is not explicitly analysable. Note that whilst we have not made any attempts to optimise the KPCA calculations above, retaining all computationally non-zero eigenvalues, the development of algorithms to improve the performance of KPCA on large data sets (e.g. $N > 3000$) has already been addressed by several authors [10, 18], as has the question of constructing reduced-set approximations to the exact KPCA results [14].

References

[1] C. J. C. Burges. Geometry and invariance in kernel based methods. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 89–116. MIT Press, Cambridge, MA, 1999.

[2] D. Butcher, T. Cootes, P. Courtney, M. Gill, and G. J. Lithgow. Model-based image analysis of a model organism for life science research. In *Proceedings of the IEE 7th International Conference on Image Processing and its Applications*, pages 392–396. 1999.

[3] T. F. Cootes, A. Hill, and C. J. Taylor. The use of Active Shape Models for locating structure in medical images. *Image and Vision Computing*, 12(6):355–366, 1994.

[4] T. F. Cootes and C. J. Taylor. Statistical models of appearance for computer vision. online document, Postscript (80 pages, 1.6Mb) at: **www.isbe.man.ac.uk/~bim/Models/app_model.ps.gz**.

[5] T. F. Cootes and C. J. Taylor. A mixture model for representing shape variation. *Image and Vision Computing*, 17(8):567–573, 1999.

[6] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active Shape Models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[7] T. Heap and D. Hogg. Extending the point distribution model using polar coordinates. *Image and Vision Computing*, 14(8):589–599, 1996.

[8] I. T. Joliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer-Verlag, New York, 1986.

[9] S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and G. Rätsch. Kernel PCA and de-noising in feature spaces. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11, pages 536–542. MIT Press, Cambridge, MA, 1999.

[10] P. Moerland. *Mixture Models for Unsupervised and Supervised Learning*. PhD thesis, Swiss Federal Institute of Technology, Lausanne, 2000. Available as IDIAP Research Report IDIAP-RR 00-18.

[11] S. Romdhani, S. Gong, and A. Psarrou. A multi-view nonlinear active shape model using kernel PCA. In T. Pridmore and D. Elliman, editors, *Proceedings of the 10th British Machine Vision Conference (BMVC99)*, pages 483–492. BMVA Press, 1999.

[12] R. Rosipal, M. Girolami, and L. J. Trejo. Kernel PCA for feature extraction and de-noising in non-linear regression. Technical Report No. 4, Department of Computing and Information Systems, University of Paisley, 2000.

[13] B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors. *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, MA, 1999.

[14] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. Smola. Input space vs. feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, 1999.

[15] B. Schölkopf, S. Mika, A. Smola, G. Rätsch, and K.-R. Müller. Kernel PCA pattern reconstruction via approximate pre-images. In L. Niklasson, M. Bodn, and T. Ziemke, editors, *Proceedings of the 8th International Conference on Artificial Neural Networks, Perspectives in Neural Computing*, pages 147–152. Springer Verlag, Berlin, 1998.

[16] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[17] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 327–352. MIT Press, Cambridge, MA, 1999.

[18] A.J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In P. Langely, editor, *Proceedings of the 17th International Conference on Machine Learning (ICML '00)*, pages 911–918. Morgan Kaufmann, San Francisco CA, 2000.

[19] P. D. Sozou, T. F. Cootes, C. J. Taylor, and E. C. Di-Mauro. A non-linear generalisation of PDMs using polynomial regression. In E. Hancock, editor, *Proceedings of the 5th British Machine Vision Conference (BMVC94), Volume II*, pages 397–406. BMVA Press, 1994. Also published in: Image and Vision Computing 13(5) 451-457, 1995.

[20] P. D. Sozou, T. F. Cootes, C. J. Taylor, E. C. DiMauro, and A. Lanitis. Non-linear point distribution modelling using a multi-layer perceptron. *Image and Vision Computing*, 15(6):457–463, 1997.