Nonlinear Shape Statistics via Kernel Spaces

Daniel Cremers, Timo Kohlberger, and Christoph Schnörr

Computer Vision, Graphics and Pattern Recognition Group
Department of Mathematics and Computer Science
University of Mannheim, 68131 Mannheim, Germany
{cremers, tiko, schnoerr}@uni-mannheim.de
http://www.cvgpr.uni-mannheim.de

Abstract. We present a novel approach for representing shape knowledge in terms of example views of 3D objects. Typically, such data sets exhibit a highly nonlinear structure with distinct clusters in the shape vector space, preventing the usual encoding by linear principal component analysis (PCA). For this reason, we propose a nonlinear Mercer-kernel PCA scheme which takes into account both the projection distance and the within-subspace distance in a high-dimensional feature space. The comparison of our approach with supervised mixture models indicates that the statistics of example views of distinct 3D objects can fairly well be learned and represented in a completely unsupervised way.

Keywords: Nonlinear shape statistics, Mercer kernels, nonlinear density estimation, shape learning, variational methods, kernel PCA

1 Introduction

One of the central questions in computer vision is how to model the link between external visual input and internally represented, previously acquired knowledge. For the case of image segmentation, prior information on the shape of expected objects can drastically improve segmentation results [9,10]. A conceptually attractive way of incorporating prior information is given by a variational approach in which external image information and statistically acquired knowledge about the shape of expected objects are combined in a single cost functional [6]:

$$E = E_{image} + E_{shape}\,. \qquad (1)$$

The present paper is concerned with the question of how to construct such a shape energy, which measures the similarity of a given shape to a set of training shapes. We focus on encoding views of distinct objects in an unsupervised way.

In most of the models of shape variability it is assumed that the training shapes define some linear subspace of the shape space [4]. Though quite powerful in many applications, this assumption only has limited validity if the observed deformations are more complex. It fully breaks down once shapes of different classes are included in the training set, such as those corresponding to different objects or just different views of a single 3D object. An example is given in

B. Radig and S. Florczyk (Eds.): DAGM 2001, LNCS 2191, pp. 269–276, 2001.

Figure 1, which shows a sampling along the first principal component for a set of 10 hand shapes containing right and left hands: the assumption of a linear distribution obviously results in an unwanted mixing up of the two classes.

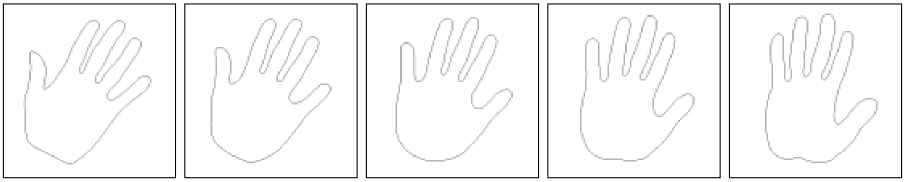

Fig. 1. Mixing of two classes in a Gaussian model: Sampling along the first principal component around the mean (center) for a training set of 10 hands, comprising both left and right hands. Shapes of different classes are *morphed* in an undesirable way.

Several approaches have been undertaken to model nonlinear shape variability. They often suffer from certain drawbacks, namely they assume some prior knowledge about the structure of the nonlinearity [8], or the number of underlying classes [3], or they involve an intricate model construction [2].

An elegant and promising way to avoid these drawbacks is to employ feature spaces induced by *Mercer kernels* [1], in order to indirectly model a nonlinear transformation $\Phi(x)$ of the original data from a space X into a potentially infinite-dimensional space Y, aspiring a simpler distribution of the mapped data in Y. The search for an appropriate nonlinearity Φ is replaced by the search for an appropriate kernel function $k(x, y)$ defining the scalar product on Y:

$$k(x, y) = (\Phi(x), \Phi(y)) \, . \tag{2}$$

With great success, this Mercer kernel approach has been used for the purpose of *classification* [5]. By contrast, our aim in the present paper is that of constructing a similarity measure by *probability density estimation*. We therefore propose to approximate the nonlinearly mapped data points $\Phi(x)$ by a Gaussian probability density *in the high-dimensional space* Y. It turns out that this can be done in the framework of Mercer kernels, i.e. all nonlinearities Φ can be expressed in terms of scalar products.

The resulting nonlinear density estimate in the original space X does not assume any prior information about the number of classes. Comparison with a supervised mixture model on simulated 2D data and its application to silhouettes of various 3D objects reveals that our estimate captures the essential nonlinear structure in the original (shape) space, although being fully unsupervised.

Our method of density estimation is related to the so-called *kernel PCA*, which shall therefore be reviewed in the next section.

2 Kernel Principal Component Analysis

In [13] a method to perform nonlinear principal component analysis is proposed. This is done by assuming an appropriate nonlinear transformation $\Phi(x_i)$ of the training data $\{x_i\}_{i=1,\ldots,\ell}$ into a space Y and performing a linear principal component analysis of the transformed data in Y (after centering it in Y). It is shown that the nonlinearity Φ enters the relevant expressions only in terms of scalar products (2). Therefore the choice of an appropriate nonlinear transformation Φ corresponds to the choice of an appropriate kernel $k(x, y)$. The eigenvectors in Y can be expressed as linear combinations of the mapped training data:

$$V_k = \sum_{i=1}^{\ell} \alpha_i^k \Phi(x_i)\,, \tag{3}$$

with known coefficients α_i^k. The projection of a mapped point $\Phi(z)$ on the eigenvector V_k is therefore given by:

$$\beta_k := (V_k, \Phi(z)) = \sum_{i=1}^{\ell} \alpha_i^k k(x_i, z)\,. \tag{4}$$

In [12] this kernel PCA is applied to pattern reconstruction. To this end the authors propose to minimize the distance

$$\rho(z) = ||P_r\Phi(z) - \Phi(z)||^2 \tag{5}$$

of a mapped sample point to its projection onto the subspace spanned by the first r eigenvectors:

$$P_r\Phi(z) = \sum_{k=1}^{r} \beta_k \, V_k\,. \tag{6}$$

The distance (5) can be expressed in terms of the kernel function (2). For a suitable kernel, a corrupted pattern z is reconstructed by minimizing (5).

3 Density Estimation in Kernel Space

In the present paper we deviate from the kernel PCA formulation above, namely we propose to perform a nonlinear probability density estimation by exploiting kernel spaces. We model the statistical distribution of the *nonlinearly mapped* data by a Gaussian distribution in Y. After centering, the covariance matrix in Y is given by

$$\Sigma_\Phi := \sum_{i=1}^{\ell} \Phi(x_i)\,\Phi(x_i)^t\,. \tag{7}$$

Let $\{\lambda_i\}_{i=1,\ldots,r}$ be the nonzero eigenvalues of Σ_Φ and V the matrix containing the respective eigenvectors V_K. In general Σ_Φ is not invertible and needs to be appropriately regularized (cf. [7]), for example by replacing all zero eigenvalues by the smallest non-zero eigenvalue λ_r. The inverse of this matrix is:

$$\Sigma_\Phi^* = V \begin{pmatrix} \lambda_1^{-1} & & & \\ & \lambda_2^{-1} & & \\ & & \ddots & \\ & & & \lambda_r^{-1} \end{pmatrix} V^t \; + \; \lambda_r^{-1} \cdot \left(I \; - \; V\,V^t\right)\,. \tag{8}$$

Approximating the distribution of mapped points in Y by a Gaussian density

$$\mathcal{P}(z) \propto \exp\left(-\frac{1}{2}\Phi(z)^t \, \Sigma_\Phi^* \, \Phi(z)\right), \tag{9}$$

corresponds (up to scaling) to an energy of the form:

$$E(z) = \Phi(z)^t \, \Sigma_\Phi^* \, \Phi(z) \,. \tag{10}$$

Using definition (8), the energy is split into two terms:

$$E(z) = \sum_{k=1}^{r} \lambda_k^{-1} \, (V_k, \Phi(z))^2 \quad + \quad \lambda_r^{-1} \left(|\Phi(z)|^2 - \sum_{k=1}^{r} (V_k, \Phi(z))^2 \right) . \tag{11}$$

Inserting expansion (3) of the eigenvectors V_k and the kernel (2) we get:

$$E(z) = \sum_{k=1}^{r} \left(\sum_{i=1}^{\ell} \alpha_i^k \, k(x_i, z) \right)^2 \cdot \left(\lambda_k^{-1} - \lambda_r^{-1}\right) + \lambda_r^{-1} \cdot k(z,z) \,. \tag{12}$$

Again, the nonlinearity Φ only appears in terms of the kernel function. Starting from a shape vector z, minimization of (12) increases its similarity to the training data $\{x_i\}$.

How and why does energy (10) differ from distance (5) proposed in [12]? The second term in (11), weighted by λ_r^{-1}, is identical with (5). It corresponds to the distance of a mapped point $\Phi(z)$ to the feature space F, which is the subspace of Y spanned by the mapped training data. Following an analogous derivation in the linear setting [11], we call this term *distance from feature space* (DFFS). The first term in (11) is called *distance in feature space* (DIFS). Both of these distances are visualized in Figure 2: the original data is mapped from the space $\mathbb{R}^n$ to a (generally higher dimensional) space Y by the nonlinear mapping Φ. The space Y is the direct sum of F and its orthogonal complement $\overline{F}$ in Y.

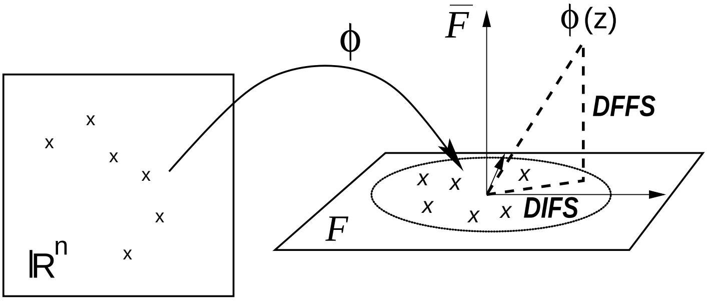

Fig. 2. Nonlinear mapping into $Y = F \bigoplus \overline{F}$ and the distances DIFS and DFFS.

In order to measure how similar a point z is to the training data $\{x_i\}$, both distances – DIFS *and* DFFS – need to be included. The DFFS by itself is not sufficient: it completely ignores how the mapped training data is distributed in F. Moreover, one can easily imagine the mapped test point $\Phi(z)$ to be far away from the mapped training data, while still being at exactly the same DFFS.

Including the DIFS as proposed in (11) accounts for the distance of the projection $P_r\Phi(z)$ *within* F from the mapped training data $\{\Phi(x_i)\}_{i=1,\dots,\ell}$. It is the Mahalanobis distance in the feature subspace F. Therefore, (11) is a more reliable measure of the similarity of a test point z to the training data $\{x_i\}$.

4 Numerical Results

4.1 Unsupervised Density Estimation via Kernel Spaces versus Supervised Mixture Models

Given the information which class each training point belongs to, one can construct a mixture model of Gaussian distributions as a nonlinear extension of PCA. For each class i one calculates mean m_i and covariance matrix Σ_i. The total probability is the sum of the probabilities for each class. The corresponding energy is given by:

$$E(z) = -\frac{1}{\beta} \log \left[\sum_i c_i \, \exp(-\beta E_i(z)) \right] \,, \quad \text{where } c_i := |2\pi\Sigma_i|^{-1/2} \tag{13}$$

and

$$E_i(z) = \frac{1}{2}(z - m_i)^t \, \Sigma_i^{-1} \, (z - m_i) \,. \tag{14}$$

The additional parameter β is introduced to allow smoothing. For small values of β one obtains the weighted sum of the single class energies (14):

$$E(z) \approx \frac{1}{\sum_i c_i} \sum_i c_i E_i(z) \, + \text{const} \quad \text{for } \beta \ll 1 \,. \tag{15}$$

The limit $\beta \to \infty$ gives their minimum: $\lim_{\beta\to\infty} E(z) = \min_i E_i(z) + \text{const}$.

We compared our approach (12) for a Gaussian radial basis function kernel[1]

$$k(x, y) = \exp\left(-\frac{||x - y||^2}{2\sigma^2} \right) \tag{16}$$

to the supervised case (13) on an artificial training set of 2D points, which were sampled from three different Gaussian distributions. The training data and the level-lines of the respective energies are depicted in Figure 3.

The comparison shows several advantages of our method. The kernel space approach is *unsupervised*: The class membership of a training point is neither known, nor determined beforehand. Even the knowledge that the data of each class is sampled from a Gaussian distribution is not taken into account. Yet, the qualitative comparison shows that the data distribution is approximated better than by the mixture model, which is based on the valid assumption of Gaussian distributions and which does imply the knowledge about the class membership of each point. Accordingly, the density estimate obtained by the mixture model is always restricted to ellipse-like level lines.

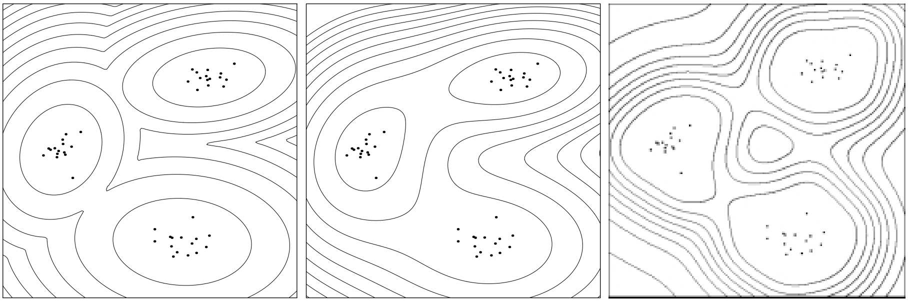

Fig. 3. Level-lines of the energies corresponding to a supervised mixture model (13) for $\beta = 1$ (**left**) and $\beta = 0.02$ (**center**) and the unsupervised density estimate via kernel spaces (10) for $\sigma = 1.5$ (**right**). These figures illustrate that our approach captures nonlinear data distributions without the need to classify the training data beforehand.

4.2 Nonlinear Shape Statistics in Kernel Space

In order to apply our distance measure (10) to realistic shapes, we parameterized the silhouettes of binarized training objects by closed spline curves. The spline curves were aligned with respect to Euclidean transformations and cyclic renumbering of the control points – see Figure 4. We used 100 control points in order to assure a sufficiently detailed contour description. The control point vectors were then used as training data to construct the energy (12), again using the kernel (16). In order to visualize the energy we projected the control point vectors of the training contours onto the first two principal components of a linear PCA[1]. The data points and the respective level lines of energy (12) are shown in Figure 5. The projection shows that our density estimate works well even in higher dimensions[2] and for distributions which are not necessarily

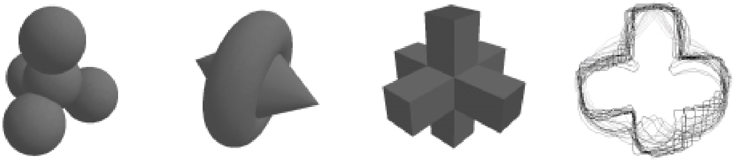

Fig. 4. 3D sample objects, and aligned silhouettes for several views of these objects. Applying linear PCA to the training set on the right would not produce an accurate description of the shape variability.

[1] Note that linear PCA is only used as a coordinate frame for *visualization* of the high-dimensional data!

[2] Due to the 2D projection, Figure 5 is merely a *crude visualization* of how the data distribution is approximated in the original 200-dimensional space.

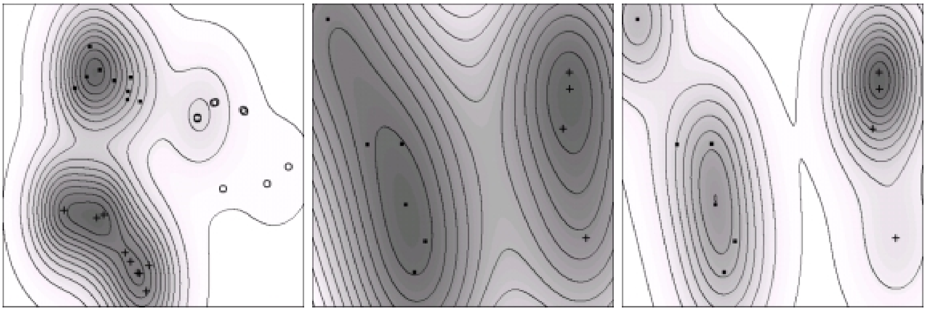

Fig. 5. Training shapes and level lines corresponding to the shape density estimate in kernel space (12), projected onto the first two principal components of a linear PCA. **Left:** Different views of objects 1 (○), 2 (+) and 3 (•) in Figure 4 for $\sigma = 0.04$. **Center:** Left hands (+) and right hands (•) (used in Figure 1) for $\sigma = 0.1$. **Right:** Hands for $\sigma = 0.04$. Clusters in high-dimensional shape space are estimated in variable detail.

Gaussian – see Figure 5. Compared to linear PCA (elliptical level lines) the true data distribution is approximated much better. This is crucial since the different shapes can be quite similar – see Figure 4, right side. Moreover, the construction of the shape energy is fully unsupervised, i.e. it does not involve the number of objects nor the number of clusters, in which the different views of one object can be separated. By changing the parameter σ in (16), one can choose how detailed the approximation of the data should be – see Figure 5, middle and right.

Note that we are *not* interested in *classification* of the objects, we merely want a measure of how *similar* an object is to a set of training objects given their 2D projections. It is therefore irrelevant whether all projections of *one 3D object* can be associated with *one cluster*. Rather we expect to obtain several clusters corresponding to the stable views of each object.

5 Conclusion

We presented a method to perform nonlinear density estimation in the framework of kernel spaces. A set of training points is mapped to a higher dimensional space Y by a nonlinear mapping Φ. The distribution of mapped points is then approximated by a Gaussian distribution in Y. Back projection to the original space allows a visualization of the estimated density. Comparison to supervised mixture models shows the advantages of our approach – namely that it is fully unsupervised and that the data distribution is approximated more appropriately. An application of this density estimation to silhouettes of 3D objects shows that the density estimate via kernel spaces seems to be well suited for high-dimensional and highly nonlinear data distributions. We argued that the distance measure corresponding to the density estimation in kernel spaces is more reliable than that obtained in kernel PCA [12].

Ongoing work focuses on ways to automatically estimate the optimal size of the parameter σ and on the application of the proposed density estimation to image segmentation [6].

Acknowledgments. We thank H. G. Schuster, J. C. Claussen, C. Kervrann, J. Keuchel, H.H. Bülthoff and J. Denzler for stimulating discussions.

References

1. B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proc. of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, PA, 1992. ACM Press.
2. B. Chalmond and S. C. Girard. Nonlinear modeling of scattered multivariate data and its application to shape change. *IEEE Trans. Patt. Anal. Mach. Intell.*, 21(5):422–432, 1999.
3. T.F. Cootes and C.J. Taylor. A mixture model for representing shape variation. *Image and Vis. Comp.*, 17(8):567–574, 1999.
4. T.F. Cootes, C.J. Taylor, D.M. Cooper, and J. Graham. Active shape models – their training and application. *Comp. Vision Image Underst.*, 61(1):38–59, 1995.
5. V. Cortes, C. and Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.
6. D. Cremers, C. Schnörr, and J. Weickert. Diffusion–snakes: Combining statistical shape knowledge and image information in a variational framework. In *IEEE Workshop on Variational and Level Set Methods*, Vancouver, Canada, Jul. 13, 2001. To appear.
7. D. Cremers, C. Schnörr, J. Weickert, and C. Schellewald. Diffusion–snakes using statistical shape knowledge. In G. Sommer and Y.Y. Zeevi, editors, *Algebraic Frames for the Perception-Action Cycle*, volume 1888 of *Lect. Not. Comp. Sci.*, pages 164–174, Kiel, Germany, Sept. 10–11, 2000. Springer.
8. T. Heap and D. Hogg. Automated pivot location for the cartesian-polar hybrid point distribution model. In *Brit. Machine Vision Conference*, pages 97–106, Edinburgh, UK, Sept. 1996.
9. C. Kervrann and F. Heitz. A hierarchical markov modeling approach for the segmentation and tracking of deformable shapes. *Graphical Models and Image Processing*, 60:173–195, 5 1998.
10. M.E. Leventon, W.E.L. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *Proc. Conf. Computer Vis. and Pattern Recog.*, volume 1, pages 316–323, Hilton Head Island, South Carolina, June 13–15, 2000.
11. B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 19(7):696–710, 1997.
12. B. Schölkopf, S. Mika, Smola A., G. Rätsch, and Müller K.-R. Kernel PCA pattern reconstruction via approximate pre-images. In L. Niklasson, M. Boden, and T. Ziemke, editors, *Internat. Conf. on Art. Neural Networks ICANN*, pages 147–152, Berlin, Germany, 1998. Springer.
13. B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.