Statistical Shape Analysis using Kernel PCA

Yogesh Rathi   Samuel Dambreville   Allen Tannenbaum

School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, Georgia 30332-0250

ABSTRACT

Mercer kernels are used for a wide range of image and signal processing tasks like de-noising, clustering, discriminant analysis etc. These algorithms construct their solutions in terms of the expansions in a high-dimensional feature space F. However, many applications like kernel PCA (principal component analysis) can be used more effectively if a pre-image of the projection in the feature space is available. In this paper, we propose a novel method to reconstruct a unique approximate pre-image of a feature vector and apply it for statistical shape analysis. We provide some experimental results to demonstrate the advantages of kernel PCA over linear PCA for shape learning, which include, but are not limited to, ability to learn and distinguish multiple geometries of shapes and robustness to occlusions.

Keywords: kernel PCA, Shape Analysis, Statistics on Shape, PCA, Pre-image

1. INTRODUCTION

Kernel methods, in particular, kernel PCA has been the focus of research in the pattern recognition community.[1,2] The basic idea behind these methods is to map the data in the input space $\mathbf{x} \in \chi$ to a feature space F via some nonlinear map ϕ, and then apply a linear method in F to do further analysis. Kernel PCA[3] is a nonlinear feature extractor, where PCA is performed in the feature space F which is equivalent to doing nonlinear PCA in the input space χ. While the mapping ϕ from input space to feature space is of primary importance in kernel based methods, the reverse mapping from the feature space to input space can also be quite useful. For example, in image (signal) denoising, a given set of noisy images can be denoised by applying kernel PCA. The problem however is that the denoised image so obtained is in the feature space and not in the input space. Hence, the need to obtain the pre-image (in the input space χ) of the projection of a feature vector given in F. Another example is in visualizing the solution of a kernel-based clustering algorithm. In this work, we use kernel PCA for doing statistical shape analysis on a set of shapes embedded in a signed distance function.[4] Knowing the pre-image can be quite useful in this case to visualize the different modes of variation of the shapes in the training set. It can also be used to provide shape prior for various tracking and segmentation tasks.

As demonstrated by Mika,[1] the exact pre-image typically does not exist and one can only settle for an approximate solution. But even this may be non-trivial as the dimensionality of the feature space can be infinite. For certain invertible kernels, this nonlinear problem can be solved using a fixed-point iteration method as proposed by Scholkopf and Mika.[1,3] However, this method is dependent on the initial starting point and is highly susceptible to local minima. To circumvent this problem, a new algorithm[2] was proposed to uniquely reconstruct the pre-image by utilizing distances of vectors in feature space and its relation to multi-dimensional scaling (MDS). In this method, to reconstruct the pre-image of a test vector $\mathbf{x} \in \chi$, the authors first find the distance of this point to all the training points in the feature space. They then find a new coordinate system to represent the data in the input space by performing SVD (singular value decomposition) on a matrix of n nearest neighbors and then use MDS to project the solution into this new coordinate system. The eigenvectors of this new coordinate system are then used to find the approximate pre-image in the original input space. Even though the method uses a couple of approximations for reconstruction, nevertheless, the results are impressive.

Further author information: (Send correspondence to Yogesh Rathi)
Yogesh Rathi: E-mail: yogesh.rathi@bme.gatech.edu, Telephone: 1 404 385 5062
Samuel Dambreville: E-mail: samuel.dambreville@bme.gatech.edu, Telephone: 1 404 385 5062
Allen Tannenbaum: E-mail: tannenba@ece.gatech.edu, Telephone 1 404 894 7582

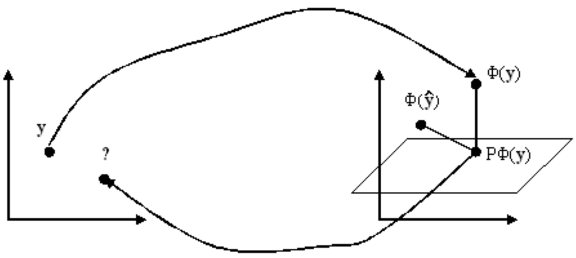


Figure 1. The pre-image problem in kernel PCA

2. OUR CONTRIBUTION

In the present work, we formulate the problem so that the reconstruction algorithm does not use MDS and only uses the distances in the feature space to find an approximate pre-image. The goal is to find the pre-image of the projection (in the kernel PCA space) of a test point $\mathbf{y} \in \chi$, as shown in Figure 1. The reconstructed pre-image $\hat{\mathbf{y}}$ satisfies the necessary condition for minimizing the distance between the feature point $\phi(\hat{\mathbf{y}})$ and the projection in the PCA space $P\phi(\mathbf{y})$, i.e., $min \parallel \phi(\hat{\mathbf{y}}) - P\phi(\mathbf{y}) \parallel^2$. Since the method performs only one approximation in the feature space, it improves accuracy and reduces computation time. The algorithm is then used to perform statistics on shapes by representing each shape as the zero level set of a signed distance function.[4] A comparison with linear PCA[4] shows the advantages of using kernel PCA for shape analysis.

A further improvement is made by noting the fact that the L_2 norm is not the best metric to use to find distances between shapes. Hence the kernel function is modified to use distances (similarities) between shapes instead of the L_2 norm. We also show how to perform reconstruction with not just the Gaussian kernel but also with a polynomial kernel.

The rest of the paper is organized as follows: The next section gives a brief overview of the kernel PCA method, section 4 demonstrates the reconstruction algorithm while in section 5 we describe certain experiments to test the proposed algorithm and section 6 gives future research directions.

3. KERNEL PCA

In this section, we briefly review the concept of kernel PCA as described by Kwok *et. al.*[2] Let $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$ be a set of training data in the input space $\chi = \mathbf{R}^d$. Kernel PCA performs the traditional linear PCA in the feature space corresponding to the kernel $k(.,.)$. The kernel gives the inner product between two points in the feature space, i.e., $k(\mathbf{x}, \mathbf{y}) =< \phi(\mathbf{x}), \phi(\mathbf{y}) >$. This fact can be used to obtain the eigenvectors in the feature space F even though the non-linear map ϕ is unknown. Analogous to linear PCA, it involves the following eigen decomposition

$$HKH = U\Sigma U^T,$$

where, K is the kernel matrix with entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, H is the centering matrix given by

$$H = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T,$$

I is the $N \times N$ identity matrix, $\mathbf{1} = [11...1]^T$ is an $N \times 1$ vector, $U = [\mathbf{a}_1, ..., \mathbf{a}_N]$ with $\mathbf{a}_i = [a_{i1}, ..., a_{iN}]^T$ is the matrix containing the eigenvectors and $\Sigma = diag(\lambda_1, ..., \lambda_N)$ contains the corresponding eigenvalues. Denote the mean of the ϕ-mapped data by $\bar{\phi} = \frac{1}{N}\sum_{i=1}^{N}\phi(\mathbf{x}_i)$ and define the "centered" map $\tilde{\phi}$ as :

$$\tilde{\phi}(\mathbf{x}) = \phi(\mathbf{x}) - \bar{\phi}.$$

The kth orthonormal eigenvector of the covariance matrix in the feature space can then be shown to be[3]

$$V_k = \sum_{i=1}^{N} \frac{a_{ki}}{\sqrt{\lambda_k}} \tilde{\phi}(\mathbf{x}_i).$$

Denote the projection of the ϕ-image of a test point $\mathbf{x}$ onto the kth component by β_k. Then,

$$\beta_k = \frac{1}{\sqrt{\lambda_k}} \sum_{i=1}^{N} a_{ki} \tilde{k}(\mathbf{x}, \mathbf{x}_i), \tag{1}$$

where,

$$\begin{aligned} &\tilde{k}(\mathbf{x}, \mathbf{y}) =< \tilde{\phi}(\mathbf{x}), \tilde{\phi}(\mathbf{y}) >= k(\mathbf{x}, \mathbf{y}) - \frac{1}{N}\mathbf{1}^T\mathbf{k}_x - \frac{1}{N}\mathbf{1}^T\mathbf{k}_y + \frac{1}{N^2}\mathbf{1}^T K \mathbf{1} \\ &\text{with} \quad \mathbf{k}_x = [k(\mathbf{x}, \mathbf{x}_1), ..., k(\mathbf{x}, \mathbf{x}_N)]^T \end{aligned} \tag{2}$$

The projection of $\phi(\mathbf{x})$ onto the subspace spanned by the first n eigenvectors is given by :

$$P\phi(\mathbf{x}) = \sum_{k=1}^{n} \beta_k V_k + \bar{\phi}$$

4. THE RECONSTRUCTION ALGORITHM

There are many functions that can be potential candidates for the choice of the kernel function $k(.,.)$. The most popular choice is the exponential kernel given by:

$$k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{d^2(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2}} \tag{3}$$

where $d^2(\mathbf{x}_i, \mathbf{x}_j)$ is a distance measure in the input space. Thus, the kernel is a function of the input space distance, i.e., $k(.) = f(d^2)$. For any two points $\mathbf{x}_i$ and $\mathbf{x}_j$ in the input space, we can obtain the distance $d(\mathbf{x}_i, \mathbf{x}_j)$. Similarly, one can also compute the feature-space distance $\tilde{d}(\phi(\mathbf{x}_i), \phi(\mathbf{x}_j))$. More specifically, the squared distance between the projection of a test point onto the first n eigenvectors (in the feature space) $P\phi(\mathbf{x})$ and an element of the training set $\phi(\mathbf{x}_i)$ can be written as[2]:

$$\tilde{d}^2(\phi(\mathbf{x}_i), P\phi(\mathbf{x})) = \| \phi(\mathbf{x}_i) - P\phi(\mathbf{x}) \|^2 = \| P\phi(\mathbf{x}) \|^2 + \| \phi(\mathbf{x}_i) \|^2 - 2\, P\phi(\mathbf{x})^T \phi(\mathbf{x}_i)$$

After some matrix manipulations, this squared distance can be written only in terms of the kernel as:

$$\tilde{d}^2(\phi(\mathbf{x}_i), P\phi(\mathbf{x})) = \left(\mathbf{k}_x + \frac{1}{N}K\mathbf{1} - 2\mathbf{k}_{x_i}\right)^T H^T M H \left(\mathbf{k}_x - \frac{1}{N}K\mathbf{1}\right) + \frac{1}{N^2}\mathbf{1}^T K \mathbf{1} + K_{ii} - \frac{2}{N}\mathbf{1}^T\mathbf{k}_{x_i} \tag{4}$$

where $M = \sum_{k=1}^{n} \frac{1}{\lambda_k} \mathbf{a}_k\, \mathbf{a}_k^T$ and $K_{ii} = k(\mathbf{x}_i, \mathbf{x}_i)$.

Moreover, for many commonly used invertible kernels, there is a simple relationship[5] between the input space distance $d_{ij}^2 = d^2(\mathbf{x}_i, \mathbf{x}_j)$ and the feature space distance $\tilde{d}_{ij}^2 = \tilde{d}^2(\phi(\mathbf{x}_i), \phi(\mathbf{x}_j))$:

$$\tilde{d}_{ij}^2 = \| \phi(\mathbf{x}_i) - \phi(\mathbf{x}_j) \|^2 = k(\mathbf{x}_i, \mathbf{x}_i) + k(\mathbf{x}_j, \mathbf{x}_j) - 2k(\mathbf{x}_i, \mathbf{x}_j) = K_{ii} + K_{jj} - 2f(d_{ij}^2) \tag{5}$$

Thus, for an invertible kernel we get

$$f(d_{ij}^2) = \frac{1}{2}(K_{ii} + K_{jj} - \tilde{d}_{ij}^2) \Longrightarrow d_{ij}^2 = f^{-1}\left(\frac{1}{2}(K_{ii} + K_{jj} - \tilde{d}_{ij}^2)\right) \tag{6}$$

Our goal is to reconstruct the pre-image $\hat{\mathbf{x}}$ in the input space of the projection $P\phi(\mathbf{x})$. In general, however, the pre-image may not exist. Hence, we are only interested in finding the approximate pre-image of the projection $P\phi(\mathbf{x})$. This can be achieved by minimizing the following error:

$$\rho(\hat{\mathbf{x}}) = \| \phi(\hat{\mathbf{x}}) - P\phi(\mathbf{x}) \|^2$$

The extremum can be obtained by setting $\nabla_{\hat{\mathbf{x}}}\rho = 0$. In what follows, we derive an expression for reconstructing the pre-image of the projection using different kernels and different types of distances d^2 in the input space.

4.1. Exponential Kernel

If the kernel function k is given by the exponential kernel (3) and if we assume that the distance in the input space is given by the Euclidean distance $d^2(\mathbf{x}_i, \mathbf{x}_j) = \| \mathbf{x}_i - \mathbf{x}_j \|^2$, then setting $\nabla_{\hat{\mathbf{x}}}\rho = 0$ gives:

$$\hat{\mathbf{x}} = \frac{\sum_{i=1}^{n} \tilde{\gamma}_i \exp\left(-\| \hat{\mathbf{x}} - \mathbf{x}_i \|^2/(2\sigma^2)\right) \mathbf{x}_i}{\sum_{i=1}^{n} \tilde{\gamma}_i \exp\left(-\| \hat{\mathbf{x}} - \mathbf{x}_i \|^2/(2\sigma^2)\right)} \tag{7}$$

where $\gamma_i = \sum_{k=1}^{n} \beta_k a_{ki}$ and $\tilde{\gamma}_i = \gamma_i + \frac{1}{N}(1 - \sum_{j=1}^{N} \gamma_j)$. This is the expression that has been used in the fixed point iteration scheme proposed by Mika *et. al.*[1] As can be easily seen, the pre-image in this case will depend on the starting point and is likely to get stuck in local minima. A unique pre-image can however be obtained by using equation (6) as follows:

$$\begin{aligned} \hat{\mathbf{x}} = \frac{\sum_{i=1}^{n} \tilde{\gamma}_i \exp\left(-d^2(\hat{\mathbf{x}}, \mathbf{x}_i)/(2\sigma^2)\right) \mathbf{x}_i}{\sum_{i=1}^{n} \tilde{\gamma}_i \exp\left(-d^2(\hat{\mathbf{x}}, \mathbf{x}_i)/(2\sigma^2)\right)} &= \frac{\sum_{i=1}^{n} \tilde{\gamma}_i \left(\frac{1}{2}(2 - \tilde{d}^2(\phi(\hat{\mathbf{x}}), \phi(\mathbf{x}_i))\right) \mathbf{x}_i}{\sum_{i=1}^{n} \tilde{\gamma}_i \left(\frac{1}{2}(2 - \tilde{d}^2(\phi(\hat{\mathbf{x}}), \phi(\mathbf{x}_i))\right)} \\ &\approx \frac{\sum_{i=1}^{n} \tilde{\gamma}_i \left(\frac{1}{2}(2 - \tilde{d}^2(P\phi(\mathbf{x}), \phi(\mathbf{x}_i))\right) \mathbf{x}_i}{\sum_{i=1}^{n} \tilde{\gamma}_i \left(\frac{1}{2}(2 - \tilde{d}^2(P\phi(\mathbf{x}), \phi(\mathbf{x}_i))\right)} \end{aligned} \tag{8}$$

where we have used the approximation $P\phi(\mathbf{x}) \approx \phi(\hat{\mathbf{x}})$ (assuming $\rho(\hat{\mathbf{x}}) \approx 0$) and the relation $d^2(\hat{\mathbf{x}}, \mathbf{x}_i) = -2\sigma^2 \log\{\frac{1}{2}(K_{ii} + K_{jj} - \tilde{d}^2(\phi(\hat{\mathbf{x}}), \phi(\mathbf{x}_i))\}$ obtained from equation (6). Thus, a unique pre-image can now be obtained using simple algebraic calculations in a single step (no iteration is required). This is one of main contributions of this paper. It should also be noted that, the proposed method is quite different from the one proposed by Kwok[2] which uses multi-dimensional scaling (MDS) and an expensive computation of singular value decomposition (SVD) in the input space to obtain the co-ordinates of the pre-image. Thus, our method not only reduces computation time, but also is more accurate since it does not require another level of approximation (projecting back in the input co-ordinate system) to compute the pre-image. Another advantage of the proposed method is that it can be used in almost all scenarios, whereas in the method by Kwok,[2] computation of SVD may not be possible if the dimension of the input vector $\mathbf{x}$ is large (for eg., an image of very large dimensions).

4.1.1. Reconstruction with shape metrics

Geometric active contours (GAC) have been widely used for such tasks as segmentation and tracking.[4,6] In this methodology, a closed curve (shape) is embedded as the zero level set of a signed distance function. An initial guess of the contour is then evolved to minimize an image based energy functional. However, only image based information is quite insufficient to segment objects if the image contains noise, clutter or occlusion.[4] Hence, prior knowledge about the possible space of shapes can be quite useful in such tasks. One of the pioneering works in this direction was done by Leventon[4] where a shape prior term was also used in addition to the image based term in the curve evolution process. The shape prior was obtained by doing linear PCA on a set of signed distance functions. Linear PCA assumes a Gaussian distribution for the variations in shape (in the training data set). In reality however, the distribution could be more complex and is better captured using a nonlinear method like kernel PCA.[3]

In the rest of the paper, we assume that the input space consists of a contour (shape) embedded in a signed distance function. Thus, we have $\mathbf{x} \in \chi = \mathbf{R}^{N^2}$, for a shape represented on a discrete grid of size $N \times N$. As can be clearly seen (Figure 3), the Euclidean distance $d^2(\mathbf{x}_i, \mathbf{x}_j) = \| \mathbf{x}_i - \mathbf{x}_j \|^2$ is not the best measure to find distance between shapes (assuming $\mathbf{x}_i, \mathbf{x}_j$ are registered using a suitable method[6]). Hence, we now derive the expression for reconstructing the pre-image using distances between shapes. One measure of similarity between shapes is given by[7]:

$$d^2(\mathbf{x}_i, \mathbf{x}_j) = \int\int_{\Omega} (H\mathbf{x}_i - H\mathbf{x}_j)^2 \, dA, \quad \text{with} \quad H\mathbf{x} = \begin{cases} 1 & \mathbf{x} \geq 0 \,, \\ 0 & \text{else} \,. \end{cases} \tag{9}$$

where $H(\mathbf{x})$ is the well-known heavy side function. This gives a metric in the space of shapes and can be easily used in the current framework. Using this metric, and setting $\nabla_{\hat{\mathbf{x}}}\rho = 0$, we get the following expression for reconstructing the pre-image (for the exponential kernel) :

$$H\hat{\mathbf{x}} = \frac{\sum_{i=1}^{n} \tilde{\gamma}_i \left(\frac{1}{2}(2 - \tilde{d}^2(P\phi(\mathbf{x}), \phi(\mathbf{x}_i))\right) H\mathbf{x}_i}{\sum_{i=1}^{n} \tilde{\gamma}_i \left(\frac{1}{2}(2 - \tilde{d}^2(P\phi(\mathbf{x}), \phi(\mathbf{x}_i))\right)} \tag{10}$$

Of course, the pre-image looks like a binary map (1 inside, 0 outside) on a 2D grid, however it is fairly easy to find the actual object contour (points that lie on the 0.5 level set). A different measure of similarity between shapes is given by[8]:

$$d^2(\mathbf{x}_i, \mathbf{x}_j) = \int_{p \in Z(\mathbf{x}_i)} EDT_{\mathbf{x}_j}(p)dp + \int_{p \in Z(\mathbf{x}_j)} EDT_{\mathbf{x}_i}(p)dp, \tag{11}$$

where $EDT_{\mathbf{x}_i}$ is the Euclidean distance function of the zero level set of $\mathbf{x}_i$ (one can think of it as the absolute value of $\mathbf{x}_i$), and $Z(\mathbf{x}_i)$ is the zero level set of $\mathbf{x}_i$. This distance measure is not a metric on the space of shapes, but allows for partial shape matching and hence quite useful in applications where partial shape matching is required. Also note that in this case, d^2 is not differentiable and hence a formulation similar to the one derived in equation 10 is not possible. However, an empirical formula can be obtained by using the distance measure (11) to calculate distances in input space while using equation (8) for reconstruction. Experiments (see Figure 3) show that using this distance measure in equation (8) gives slightly better results as compared to any of the distance measures discussed before.

4.2. Polynomial Kernel

In certain cases, an exponential kernel may not be the best choice, and a polynomial kernel may be used. We define the polynomial kernel by $k(\mathbf{x}, \mathbf{y}) = <\mathbf{x}, \mathbf{y}>^d = (\mathbf{x}^T\mathbf{y})^d$, where d is the degree (odd) of the polynomial. Once again, setting $\nabla_{\hat{\mathbf{x}}}\rho = 0$ gives the following expression for obtaining the pre-image:

$$\hat{\mathbf{x}} = \sum_{i=1}^{n} \tilde{\gamma}_i \left(\frac{<\hat{\mathbf{x}}, \mathbf{x}_i>}{<\hat{\mathbf{x}}, \hat{\mathbf{x}}>}\right)^{d-1} \mathbf{x}_i.$$

After some algebraic manipulations, and using the approximation $\phi(\hat{\mathbf{x}}) \approx P\phi(\mathbf{x})$, we get the following expression to reconstruct the pre-image of the projection:

$$\hat{\mathbf{x}} = \sum_{i=1}^{n} \tilde{\gamma}_i \left(\frac{\| P\phi(\mathbf{x}) \|^2 + k(\mathbf{x}_i, \mathbf{x}_i) - \tilde{d}^2(P\phi(\mathbf{x}), \mathbf{x}_i)}{2\| P\phi(\mathbf{x}) \|^2}\right)^{\frac{d-1}{d}} \mathbf{x}_i \tag{12}$$

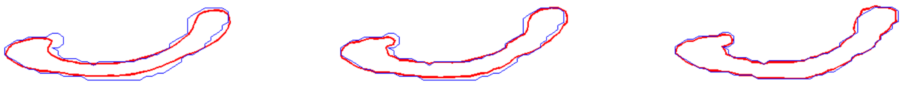
where, $\| P\phi(\mathbf{x}) \|^2$ can be easily calculated in terms of the kernel using[2]:

$$\| P\phi(\mathbf{x}) \|^2 = \left(\mathbf{k}_x + \frac{1}{N}K\mathbf{1}\right)^T H^T M H \left(\mathbf{k}_x - \frac{1}{N}K\mathbf{1}\right) + \frac{1}{N^2}\mathbf{1}^T K\mathbf{1}.$$

It should be noted that, the error in reconstruction of the pre-image is proportional to the error in assuming $\phi(\hat{\mathbf{x}}) = P\phi(\mathbf{x})$, which is very small considering the fact that we want the pre-image of the projection $P\phi(\mathbf{x})$. Also note that, if we take $d = 1$, we get the expression for doing linear PCA.

Using the same methodology, one can obtain an expression for computing the pre-image with any invertible kernel and using any type of metric in the input space. In the next section, we perform experiments to evaluate the proposed algorithm. In particular, we use it do statistical shape analysis on a set of signed distance functions.

Figure 2. Samples from the unaligned training set for corpus callosum



(a) Euclidean Distance (b) using equation (9) (c) using equation (11)

Figure 3. Reconstruction of pre-image using different distance functions. Dark curve is the pre-image, while the lighter curve is the original curve.

5. EXPERIMENTS

Figure 2 shows some samples from the training set of corpus callosum. Binary images are shown for better visualization of the variations in shape. For our purposes, we represent a curve as the zero level set of a signed distance function. The distance map can be considered as a point in a higher dimensional space ($\mathbf{x} \in \mathbb{R}^{N^2}$, where $\mathbf{x}$ was originally an $N \times N$ matrix).

Kernel PCA is performed on this training set and the proposed algorithm is then used to reconstruct the pre-image of the projection of a test point $\mathbf{x} \in \mathbb{R}^{N^2}$. The training set for the corpus callosum consisted of 23 elements. Figure 3 shows the original curve and its pre-image as obtained using an exponential kernel with 3 different distance functions as mentioned before in section 4. The reconstruction was performed using $n = 10$. Of course, the choice of σ plays an important role. A typical choice for σ is the average minimum distance between two points in the training set, i.e., $\sigma^2 = c\,\frac{1}{N}\sum_{i=1}^{N} \min_{j \neq i} d^2(\mathbf{x}_i, \mathbf{x}_j)$, where c is a user defined parameter. A larger value of σ allows more "mixing" between elements of the training set, whereas a smaller value of σ uses only a few significant members in the reconstruction process. This is demonstrated in Figure 4, where the pre-image is obtained for different values of σ.

Figure 5 shows the two principal modes of variation of the shapes in the kernel PCA space. Figure 6 shows the pre-image of the projection of a partially occluded curve. The blue curve is the original data and the red curve is the pre-image of its projection. Doing the same thing for linear PCA gives poor results as shown in Figure 6. Thus, kernel PCA is quite robust to occlusions which could be used very effectively for segmentation and tracking.

Next, we demonstrate the robustness of kernel PCA in learning objects of different geometries. Figure 7 shows a set of ventricles that were added to the training set of corpus callosum. We now find the pre-images of the projection $P\phi(\mathbf{x})$ of a set of test points. As can be seen, not only can kernel PCA distinguish between clusters of shapes, it is also robust to partial occlusions (see Figure 8). This property of kernel PCA can be extremely useful in tracking (or segmenting) multiple objects with different shape geometries. Using linear PCA however, would be quite disastrous as seen in Figure 6. For this experiment, the distance measure (9) was used.

Figure 4. Reconstructed pre-images for different values of σ. (σ increasing from left to right).

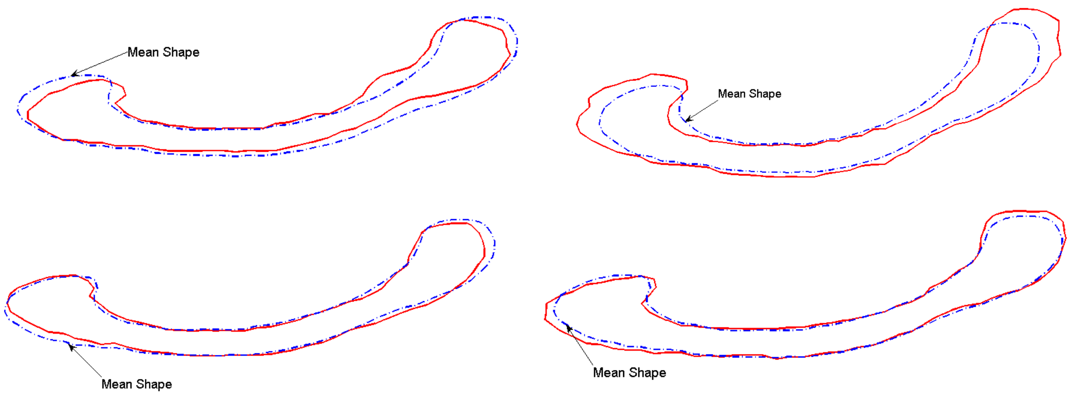


Figure 5. First 2 principal modes of variation of corpus callosum in the feature space. The 2 figures in the top row are $\mp 2\sigma$ variation of the first principal component. The second row shows the variation in the second principal direction. Dotted curves represent mean shape.

Figure 6. Figure shows the pre-image of the projection in feature space and the corresponding pre-image using linear PCA. First and third figures are obtained using kernel PCA, while the second and fourth are obtained using linear PCA.

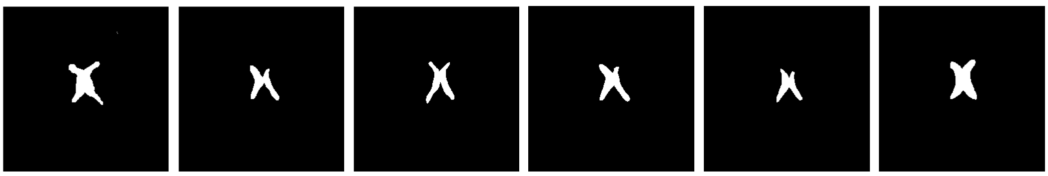


Figure 7. Samples from the unaligned training set for ventricles

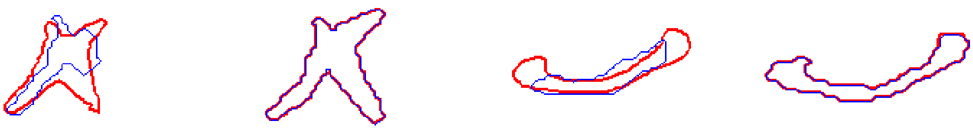


Figure 8. Figure shows the pre-image of the projection in feature space with the training set containing 23 corpus callosum and 20 ventricles. As can be seen, kernel PCA can easily distinguish between different shape geometries even if part of the original curve is occluded. The dark curve represents pre-image, while the lighter one is the original.

6. CONCLUSION

In this work, we have proposed a novel method to find the pre-image of the projection in the feature space. The method is not only computationally less complex, but also more accurate. To the best of our knowledge, this is the first time, pre-image of the projection has been used to study statistical properties of a set of training shapes. We also demonstrate the robustness of kernel PCA in learning shapes of different types irrespective of partial occlusions. The reconstructed pre-image can now be used as a shape prior for segmentation and tracking. It can also be used in many other applications like denoising images or for visualization of the solution in many clustering algorithms.

REFERENCES

1. S. Mika, B. Scholkopf, A. Smola, K.R.Muller, M.Scholz, and G.Ratsch, "Kernel pca and de-noising in feature spaces," *Advances in Neural Information Processing Systems 11.*
2. J. Kwok and I. Tsang, "The pre-image problem in kernel methods," in *20th Intl. Conference on Machine Learning*, 2003.
3. B. Scholkopf, A. Smola, and K. R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," tech. rep., Max-Planck-Institute fur biologische Kybernetik, 1996.
4. M. Leventon, E. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contours," in *Proc. CVPR*, pp. 1316–1324, IEEE, 2000.
5. B. Scholkopf, "The kernel trick for distances," *NIPS*, pp. 301–307, 2000.
6. A. Tsai, A. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, E. Grimson, and A. Willsky, "A shape-based approach to the sementation of medical imagery using level sets," *IEEE Trans. on Medical Imaging* **22**, pp. 137–153, 2003.
7. T. Chan and W. Zhu, "Level set based shape prior segmentation," tech. rep., Computational Applied Mathematics, UCLA, 2003.
8. T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin, "Modeling by example," in *ACM Transactions on Graphics (SIGGRAPH 2004)*, 2004.