Connecting the Out-of-Sample and Pre-Image Problems in Kernel Methods*

Pablo Arias
Universidad de la República
parias@fing.edu.uy

Gregory Randall
Universidad de la República
randall@fing.edu.uy

Guillermo Sapiro
University of Minnesota
guille@umn.edu

Abstract

Kernel methods have been widely studied in the field of pattern recognition. These methods implicitly map, "the kernel trick," the data into a space which is more appropriate for analysis. Many manifold learning and dimensionality reduction techniques are simply kernel methods for which the mapping is explicitly computed. In such cases, two problems related with the mapping arise: The out-of-sample extension and the pre-image computation. In this paper we propose a new pre-image method based on the Nyström formulation for the out-of-sample extension, showing the connections between both problems. We also address the importance of normalization in the feature space, which has been ignored by standard pre-image algorithms. As an example, we apply these ideas to the Gaussian kernel, and relate our approach to other popular pre-image methods. Finally, we show the application of these techniques in the study of dynamic shapes.

1. Introduction

Kernel methods have been shown to be powerful techniques for studying non-linear data. The main idea behind these methods is to map the data into a space better suited for linear algorithms. The mapping, however, is often not explicitly computed, leading to the so called "kernel trick:" The kernel function encodes the useful information about the mapping. Kernel methods have been used in numerous image processing and computer vision applications; see for example [24] for a comprehensive review on kernel methods.

Kernel methods are closely related to manifold learning techniques such as those described in [2, 5, 10, 17, 23, 26], see [3, 4, 13] for details. The aim of these algorithms is to map the original dataset into a parameter space, usually of lower dimension. The mapping is associated with a kernel function, giving a different point of view to the manifold learning problem. As a result, new manifold learning algorithms have been developed using design techniques borrowed from the kernel methods theory, *e.g.* [28].

In general, for manifold learning techniques, the mapping is only known over the training set. This mapping needs to be extended to new input points as they come, without having to re-compute the (often expensive) whole map. This is known as the *out-of-sample* problem. In addition, after operations are performed in the mapped (*feature*) space, often the corresponding data point in the original space needs to be computed. This is known as the *pre-image* problem. While both problems are treated separately in the literature, we show in this paper that they are closely related, and in particular, the Nyström extension for the out-of-sample task can be extended to address the pre-image issue as well. We should note that most of the work in the pre-image problem has been done for the Gaussian kernel. This kernel has been widely used in the field of patter classification and also for manifold learning. In [8, 9, 25] the Gaussian kernel is used to perform kernel principal component analysis (PCA) for image de-noising and shape manifold learning, outperforming ordinary PCA. In [7] a non-parametric probability density function is learned by assuming a Normal distribution in the feature space of a Gaussian kernel.

A common approach for studying both static and dynamic data is to first learn the non-linear manifold underlying the training set, *e.g.* [1, 7, 9, 19]. The learned manifold is then used for diverse applications such as activity recognition and object tracking. In these cases, both the out-of-sample extension and the pre-image problem are central issues. The out-of-sample extension is critical to handle new data as it comes in, without the necessity to re-learn the manifold, task which is computationally expensive and performed off-line. The pre-image is critical to being able to work back in the original space, either for visualization (when computing an average, for example, or when using kernel PCA for de-noising or analysis), or for operations such as tracking in the video space.

The contribution of this paper is threefold. First, we pro-

*Work partially supported by NSF, ONR, NGA, and DARPA. PA and GR performed part of this work while visiting ECE&IMA at the UofM. We thank the authors of [1] for the data and the third reviewer for pointing out important references.

pose a new approach for addressing the pre-image problem, based on the connections with the out-of-sample extension. In particular we use the Nyström extension for this purpose. We exemplify these ideas with the Gaussian kernel, although they can be generalized to other kernels. Secondly, the proposed formulation gives insight into the understanding of the pre-image problem and into some existing pre-image algorithms. Thirdly, we carefully consider the issue of the norm in the feature space, which has been previously ignored for pre-image algorithms.

As an application of the proposed pre-image technique, we analyze dynamic shapes (DS), namely a coherent temporal sequence of shapes, such as gait sequences and lips movement while talking. One of the examples shown in this work is the upsampling of DS from lips movies, obtained via interpolation in the feature space. Using the Gaussian kernel in conjunction with an appropriate metric, such interpolation is robust to local brightness changes.

A different approach here addressed considers each whole DS as a high dimensional element, instead of treating it as a sequence of static shapes, see also [27]. We apply this idea in conjunction with the Gaussian kernel to map the DS into a lower dimensional feature space, where several tasks can be performed, such as DS averaging [20], statistical modeling of DSs, classification of activities, and reconstruction of partially occluded DSs.

The rest of the paper is organized as follows. Section 2 introduces some basics concepts about kernel methods. Section 3 focuses on kernel methods applied to manifold learning. In particular, approaches to the pre-image problem are discussed and introduced in this section. The novel connection between out-of-sample and pre-image is presented in Section 4. First numerical results are presented in Section 5. Section 6 shows the application of kernel based techniques and our proposed pre-image method to the processing of dynamic shapes. Conclusions and future work are presented in Section 7.

2. Kernel methods basics

Let $\Omega = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$ be the set of training points. The kernel is a function $k : \Omega \times \Omega \to \mathbb{R}$, such that there exist a mapping $\varphi : \Omega \to \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space and the following inner-product relationship holds

$$k(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle \qquad i, j = 1, \dots, n. \tag{1}$$

Let $K \in \mathcal{M}_{n\times n}$ be the matrix containing the kernel values, $K_{ij} = k(x_i, x_j)$. If this matrix is semidefinite positive, then k is a kernel over the set Ω [24]. A mapping satisfying the dot product property (1) can be found by the eigen-decomposition of the kernel matrix K:

$$K = U\Lambda U^T = U\Lambda^{\frac{1}{2}}(U\Lambda^{\frac{1}{2}})^T, \tag{2}$$

where U is the matrix whose columns are the eigenvectors ϕ_i, $i = 1, \dots, n$, and $\Lambda = \text{diag}(\lambda_1, \lambda_1, \dots, \lambda_n)$ is the diagonal matrix of the eigenvalues in decreasing order. If we define $\varphi(x_i)$ to be the i-th row of $U\Lambda^{\frac{1}{2}}$, and since the eigenvectors are non-negative (positive semidefinite matrix), we obtain the desired mapping:

$$\varphi(x_i) = [\sqrt{\lambda_1}\phi_1(x_i), \sqrt{\lambda_2}\phi_2(x_i), \dots, \sqrt{\lambda_n}\phi_n(x_i)]. \tag{3}$$

The kernel function can then be considered as a generalization of the dot product, and therefore it is a measure of similarity between the input points. The Hilbert space $\mathcal{H}$ is called the *feature* space. When the algorithm to be applied in the feature space uses only the corresponding dot products, only the kernel values are needed, without the need for the explicit computation of the mapping functions. This is called the *kernel trick*.

3. Kernel methods and manifold learning

The output of a manifold learning algorithm is often a representation of the set of input points in a (usually) lower dimensional space, in such a way that geometric properties of interest of the underlying manifold are maintained. This mapping is often found as the eigen-decomposition of a matrix, often called the *transition matrix*. This matrix can be viewed as a kernel matrix. As mentioned above, the out-of-sample extension and pre-image problems arise naturally in this framework.

Dimensionality reduction algorithms, such as LLE, Isomap, Laplacian Eigenmaps, are in fact kernel methods. In this paper we focus on the Gaussian kernel, as often done in the literature, *e.g.* [7, 8, 25],

$$k(x, x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}. \tag{4}$$

The value of the σ parameter is very important. In order to understand this parameter it is useful to see the Gaussian kernel as a transition matrix in a random walk (up to proper normalization) [17]. The parameter σ determines the distance of reachable neighbors in one single step of the random walk. A small σ captures better the local structure of the manifold. However if σ is too small, for a finite sample, the points become disconnected. In [7, 17] for example, σ is computed as the average of the distances to the nearest neighbor, $\sigma = \frac{1}{N}\sum_{i=1}^{N} d(x_i, x_{i,1})$, where $x_{i,1}$ is the nearest neighbor of x_i. Similarly, in [18] σ is computed as the smallest distance such that every point is at least connected with one neighbor.

3.1. The Nystrom extension

Let $x \in \mathbb{R}^d$ be a new input point not in the training set. The Nyström extension, [4], states that the j-th coordinate of the kernel mapping φ for this point can be approximated as:

$$\hat{\varphi}_j(x) = \frac{1}{\sqrt{\lambda_j}} \sum_{i=1}^{n} k(x, x_i)\phi_j(x_i) \qquad j = 1, \dots, n, \quad (5)$$

or in vector form:

$$\hat{\varphi}(x) = \frac{1}{\sqrt{\Lambda}} U^T k_x, \quad (6)$$

where $k_x = [k(x, x_1), k(x, x_2), \dots, k(x, x_n)]$, and $\frac{1}{\sqrt{\Lambda}}$ stands for $(\sqrt{\Lambda})^{-1} = \text{diag}(\frac{1}{\sqrt{\lambda_1}}, \frac{1}{\sqrt{\lambda_2}}, \dots, \frac{1}{\sqrt{\lambda_n}})$. In other words, the new point x is mapped as a weighted linear combination of the corresponding maps for the training points x_i. The weights are given, modulo normalization by the eigenvalues, by the kernel relationship $k(x, x_i)$ representing the similarity between x and x_i.

Observe that while extending the mapping, we also need to extend the kernel. This is straightforward when the kernel defined over Ω is simply a known function defined in the ambient space $\mathbb{R}^d$. In other cases the extension of the kernel is not trivial. In [4] the authors propose extensions for the kernels defined by LLE, Isomap, MDS, and Laplacian Eigenmaps. These are data driven kernels, they are functions $k_n : \mathbb{R}^d \to \mathbb{R}$ that depend on the input points.

When the data is sampled from a distribution, it has been shown [3], that the functions defined by the Nyström extension converge uniformly to the eigenfunctions of the limit of the sequence of data driven kernels, given that this limit exists and that their eigenfunctions also converge. This asymptotic property makes the Nyström extension an appealing approach for the out-of-sample extension problem. See [18] for related alternatives.

3.2. The pre-image problem

The pre-image of $\psi \in \mathcal{H}$ (feature space point) is a point $x \in \mathbb{R}^d$ (original data space), such that $\varphi(x) = \psi$. Since such a point x might not exist, the pre-image problem is ill-posed. A way to circumvent this problem is to look for an approximate pre-image, *i.e.* a point $x \in \mathbb{R}^d$ such that $\varphi(x)$ is "as close as possible" to ψ. Different optimality criteria could be used, such as

$$\text{Distance: } x = \arg\min_{x \in \mathbb{R}^d} \|\varphi(x) - \psi\|^2. \quad (7)$$

$$\text{Collinearity: } x = \arg\max_{x \in \mathbb{R}^d} \left\langle \frac{\varphi(x)}{\|\varphi(x)\|}, \frac{\psi}{\|\psi\|} \right\rangle. \quad (8)$$

Mika *et al.* [21], present an analytic solution applying the collinearity criterion for the Gaussian kernel (or other radial basis function kernel). First, they assume that ψ is a linear combination of the mappings of the training set, $\psi = \sum_{i=1}^{n} \alpha_i \varphi(x_i)$. Since the Gaussian kernel is a normalized kernel, *i.e.* $k(x, x) = 1$ for all $x \in \mathbb{R}^d$, the cost function in (8) becomes:

$$\langle \varphi(x), \sum_{i=1}^{n} \alpha_i \varphi(x_i) \rangle = \sum_{i=1}^{n} \alpha_i k(x, x_i), \quad (9)$$

where we used the dot product property of the kernel (1). The maximum can be found by taking the gradient of this expression, leading to the following expression for the optimal x:

$$x = \frac{\sum_{i=1}^{n} \alpha_i k(x, x_i) x_i}{\sum_{i=1}^{n} \alpha_i k(x, x_i)}. \quad (10)$$

This implicit equation can be solved by a fixed point iteration, but suffers from local minima and instabilities [9].

In [9] an approximation to avoid the iteration is proposed. The distance $d_{\mathcal{H}}$ between the mapped points can be computed in terms of the dot products, and therefore in terms of the kernel,

$$d_{\mathcal{H}}(\varphi(x), \varphi(x'))^2 = k(x, x) + k(x', x') - 2k(x, x'). \quad (11)$$

Since the Gaussian kernel is a normalized one, we obtain:

$$d_{\mathcal{H}}(\varphi(x), \varphi(x'))^2 = 2(1 - k(x, x')). \quad (12)$$

This equation only holds for points in $\mathcal{H}$ that have an exact pre-image, *i.e.* for the points that belong to the image of the mapping, $\varphi(\mathbb{R}^d)$. However, in [9] the authors make the assumption that $\psi \approx \varphi(x)$, and use this to estimate the kernel values of x with the input points:

$$\hat{k}(x, x_i) = \frac{1}{2}(2 - d_{\mathcal{H}}(\psi, \varphi(x_i))^2) \qquad i = 1, \dots, n. \quad (13)$$

Substituting this approximation in the iterative equation (10) leads to a direct formula (which can be considered as a first step in the iteration (10)).

In [16], the authors also use (13) to estimate the kernel values, and use these values to compute the distance in the input space between the searched pre-image x and the given training points:

$$\|x - x_i\|^2 \approx 2\sigma^2 \log(\frac{1}{2}(2 - d_{\mathcal{H}}(\psi, \varphi(x_i))^2)) \quad i = 1, \dots, n. \quad (14)$$

Finding x now reduces to a localization problem solved by standard MDS [12]. This approach is not based in any of the two optimality criteria mentioned above.

4. Pre-images via the Nyström extension

We now address the connections between the out-of-sample and the pre-image problems. Observing the optimality criteria for the approximate pre-image (Eqs. (7) and (8)), it is clear that if we know of a way of extending the mapping, we could find the pre-image optimizing the corresponding cost function. Although we do not know the exact (extended) mapping, we can approximate it with the Nyström extension. If we rewrite (7) and (8) using the Nyström extension to express $\varphi(x)$, we obtain:

$$\text{Distance: } x = \arg\min_{x\in\mathbb{R}^d} \left\| \frac{1}{\sqrt{\Lambda}} U^T k_x - \psi \right\|^2. \tag{15}$$

$$\text{Collinearity: } x = \arg\max_{x\in\mathbb{R}^d} \left\langle \frac{\frac{1}{\sqrt{\Lambda}} U^T k_x}{\left\| \frac{1}{\sqrt{\Lambda}} U^T k_x \right\|}, \frac{\psi}{\|\psi\|} \right\rangle. \tag{16}$$

Working with the exact extension, both criteria are equivalent for normalized kernels, as can be seen by expanding the squared norm in Eq. (7):

$$\|\varphi(x) - \psi\|^2 = 1 + \langle\psi,\psi\rangle - 2\langle\varphi(x),\psi\rangle. \tag{17}$$

Since ψ is constant, minimizing the left-hand side of the expression is equivalent to maximizing $\langle\varphi(x),\psi\rangle$.

This equivalence is no longer true when φ is approximated by the Nyström extension, the norm of $\hat{\varphi}(x)$ is not necessarily 1. In fact, if the searched pre-image x is outside the range of the extension, $\|\hat{\varphi}(x)\|$ tends to zero. If we compute the pre-image of a point ψ with a small norm by minimizing the distance criterion, the pre-image x tends to lie outside the range of $\hat{\varphi}$. This important lack of normalization has been ignored by the previously mentioned pre-image algorithms.

To address this problem, we modify the distance criterion by projecting ψ onto the unit sphere (normalized kernel):

$$x = \arg\min_{x\in\mathbb{R}^d} \left\| \frac{1}{\sqrt{\Lambda}} U^T k_x - \frac{\psi}{\|\psi\|} \right\|^2. \tag{18}$$

Note that for the real mapping φ this problem is equivalent to the original distance criterion (7), we are forcing x to stay in the range of $\hat{\varphi}$ without modifying the original problem.

Solving Eq. (18) for k_x (the vector formed by $k(x, x_i)$) gives an approximation for the optimal kernel vector. This is a standard least squares problem, where the solution is given (for example) by the Penrose-Moore pseudo-inverse. Since U is a unitary matrix we obtain (compare with Equation (6))

$$\hat{k}_x = U\sqrt{\Lambda}\frac{\psi}{\|\psi\|}. \tag{19}$$

This optimal estimate of k_x has an intuitive interpretation. Recall that the mapping of the training points is given by the rows of $U\sqrt{\Lambda}$. Thus, the i-th component of $\hat{k}_x$ can be expressed as

$$\hat{k}_x(i) = \left\langle \phi(x_i), \frac{\psi}{\|\psi\|} \right\rangle. \tag{20}$$

Thereby, estimating the kernel values as the dot product between the mapped points and the projection of ψ on the unit sphere is equivalent to inverting the Nyström extension (Equation (6)), showing the close connections between the out-of-sample extension problem and the pre-image problem.

We could also try to find x as the point in the input space whose kernel values are closer to the kernel vector estimate given by Eq. (19):

$$x = \arg\min_{x\in\mathbb{R}^d} \|k_x - \hat{k}_x\|^2. \tag{21}$$

This problem is not equivalent to (18). Therefore solving directly for x in Eq. (18) will yield a solution whose kernel vector will not necessarily be the closest one to $\hat{k}_x$.

In order to compare $\hat{k}_x^N$, the kernel estimated by our approach (19), and $\hat{k}_x^D$, the one proposed by [16] and [9], note that

$$\begin{aligned}\hat{k}_x^D(i) &= \frac{1}{2}(2 - d_{\mathcal{H}}(\varphi(x_i),\psi)^2) \\ &= \frac{1}{2}(2 - 1 - \langle\psi,\psi\rangle + 2\langle\varphi(x_i),\psi\rangle) \\ &= \frac{1}{2}(1 - \langle\psi,\psi\rangle) + \|\psi\|\hat{k}_x^N(i). \end{aligned} \tag{22}$$

where $\hat{k}_x^{\cdot}(i)$ denotes the i-th component of vector $\hat{k}_x^{\cdot}$. The term $\frac{1}{2}(1 - \langle\psi,\psi\rangle)$ appears due to the assumption that $\langle\psi,\psi\rangle = 1$, (wrongly) made implicitly by [9, 16] when applying the relationship between the distance $d_{\mathcal{H}}$ and the kernel given by Eq. (12). Of course if $\varphi(x) \approx \psi$ then $\|\psi\| \approx 1$. However, this is in general not true. Consider for example the simple case in which ψ is an average of some mapped points that lie in the unit sphere. Their average will not have unity norm.

Once we have estimated $k(x, x_i)$, the similarity between the pre-image x and the rest of the points x_i, we must determine x. For this purpose we could solve (21). This would yield an iterative scheme. Instead, we use the approximations proposed by [9, 16].

To summarize, by means of the Nyström extension, in this section we showed the connections between the out-of-sample and the pre-image problems. Using this plus a proper handling of the vector norms in the Hilbert space, we proposed new methods for computing the pre-image, based on the novel computation of the kernel vector k_x and the approaches presented in [9] and [16].

5. Initial experimental results: Comparing the methods

To evaluate our proposed technique, we first need to define a measure of performance for the pre-image algorithms. Let $x \in \mathbb{R}^n$ be any of the approximate pre-images of $\psi \in \mathcal{H}$. In accordance to (8), the performance is mea-

Range\ Pre-Image	**A**	**B**	**C**	**D**	**E**
$(0; 0.2)$	0	0	0	0	3
$(0.2; 0.4)$	9	11	54	20	40
$(0.4; 0.6)$	37	54	39	52	42
$(0.6; 0.8)$	34	24	33	21	15
$(0.8; \infty)$	20	20	19	7	0

(a) $d = 2$ and $r_\sigma = 1$.

Range\ Pre-Image	**A**	**B**	**C**	**D**	**E**
$(0; 0.1)$	60	65	62	65	65
$(0.1; 0.2)$	33	29	31	29	29
$(0.2; 0.3)$	4	4	4	4	5
$(0.3; 0.4)$	2	2	3	2	1
$(0.4; \infty)$	1	1	0	1	0

(b) $d = 2$ and $r_\sigma = 5$.

Range\ Pre-Image	**A**	**B**	**C**	**D**	**E**
$(0; 0.01)$	0	0	0	0	0
$(0.01; 0.02)$	5	94	23	94	92
$(0.02; 0.03)$	57	6	64	6	8
$(0.03; 0.04)$	30	0	13	0	0
$(0.04; \infty)$	8	0	0	0	0

(c) $d = 30$ and $r_\sigma = 1$.

Range\ Pre-Image	**A**	**B**	**C**	**D**	**E**
$(0; 0.005)$	0	0	0	0	0
$(0.005; 0.01)$	1	39	1	39	38
$(0.015; 0.02)$	10	60	30	60	61
$(0.02; 0.025)$	44	1	50	1	1
$(0.025; \infty)$	45	0	19	0	0

(d) $d = 30$ and $r_\sigma = 5$.

Table 1. *Results for the pre-image algorithms. Each table shows the number of trials for which the collinearity error falls into the corresponding range. The total number of runs was* $m = 100$ *(*$N = 800$*).*

sured as the collinearity error between ψ and $\hat{\varphi}(x)$ (computed using the Nyström extension):

$$e_c(x) = 1 - \left\langle \frac{\hat{\varphi}(x)}{\|\hat{\varphi}(x)\|}, \frac{\psi}{\|\psi\|} \right\rangle. \tag{23}$$

The algorithms to be compared include: **A**. Distance-based pre-image [16], **B**. Direct formula approximation of the iterative pre-image [9], **C**. Pre-image **A** with k_x computed inverting the Nyström extension as here proposed, **D**. Pre-image **B** with k_x computed inverting the Nyström extension as here proposed. We also compare against the iterative pre-image method presented by Mika *et al.* [21], denoted by pre-image **E**. In order to see if the iteration improves the results its initial value is set as the pre-image **D**.

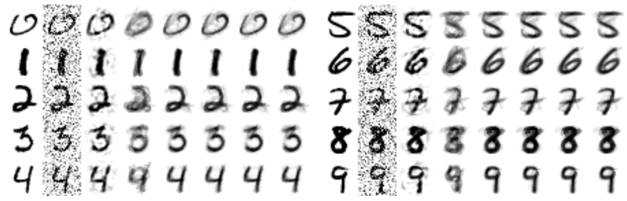
Given the set of parameters d (the input space dimension), n (the number of training points), r_σ (the number of nearest neighbors for determining σ), and m (the number of runs with $m \leqslant n$), the comparison protocol goes as follows:

1. **Initialization:** Randomly generate a set $\{x_i\}$ of n points uniformly distributed in $[0, 1]^d$, compute σ as the average distance to the r_σ-th nearest neighbors, and compute the Gaussian kernel and the mapping $\varphi(x_i)$, $i = 1, \ldots, n$.

2. **Evaluation:** For $j = 1, \ldots, m$, pick a random point $\varphi(x_j) \in \{\varphi(x_1), \varphi(x_2), \ldots, \varphi(x_n)\}$, define ψ as the mean of k nearest neighbors of $\varphi(x_j)$, compute the pre-image of ψ using each of the methods listed above, map each pre-image back into the feature space using the Nyström extension, and finally compute the collinearity error for each mapping.

Table 1 shows the comparison of the pre-images, for four sets of parameters. For high dimensions, the algorithms **B**, **D** and **E** significantly outperform the methods **A** and **C**. There is an improvement in the results obtained with the approach [16] (pre-image **A**) if the kernel estimate by (19) is used (pre-image **C**): Approximately in 75% of the trials using the Nyström "duality"-based k_x yields an improvement of about 10%. The pre-images **B** and **D** give similar results and similar to **E** for high dimensions. Except in low dimension there is almost no improvement by using the fixed point iterative pre-image **E**. The results are in general better when the dimension of the input set is high. This behavior is due to the way the σ parameter is computed. When the dimension of the input space increases, the distances between the nearest neighbors increase in relation with the size of the input set, thus the mapping becomes simpler, and thereby easier to extend [18].

We also compare the pre-image algorithms performance in image de-noising by kernel PCA [16, 21], using the 28×28 pixels MNIST handwritten digits database. The training set was built with 30 images for each digit. Given a new noisy image, de-noising is done by mapping it into the feature space, then projecting the mapped image onto the principal components of the training set, and finally computing the pre-image. Figure 1 shows the results. It is clear that pre-images **B**, **D** and **E** yield the best (and very similar) results. This is in accordance with the experiment presented in Table 1. Averaging the PSNR over the ten test digits gives the results shown in Table 1(b).



(a) De-noising handwritten digits.

A	**B**	**C**	**D**	**E**	PCA
14.62	19.74	19.31	20.48	19.50	17.17

(b) Average PSNR (dB)

Figure 1. *Kernel PCA de-noising. 1(a) Each row corresponds to a test digit. The columns are: original image, noisy image, result of linear PCA, pre-images* **A** *to* **E** *respectively. 1(b) Averaged PSNR for the test digits.*

To conclude, the new technique for estimating the kernel vector k_x represents an improvement with respect to the previous pre-image approaches. Pre-image **D** has overall good results with low computational complexity and it is theoretically founded.

6. Applications to dynamic shapes

Static shape statistics has been widely demonstrated to be crucial for many computer vision tasks, such as shape-based segmentation. When time is involved, dynamic shape priors are proving to be very important also, for applications such as video segmentation [6, 19, 22], tracking [27] and activity recognition [11].

In this section we demonstrate the applicability of the proposed kernel plus pre-image framework for learning low-dimensional representations of dynamic shapes. We exemplify with gait and lips movement data. All the pre-images shown in this section are going to be computed using pre-image $\mathbf{D}$.

Let S be a description of a shape, *e.g.* the signed distance function to the boundary of the region. In this case the L^2 distance between the signed distance functions can be used. This is just one simple example here used for illustration purposes. A different distance is used for the lips data. A dynamic shape is a sequence of shapes, $\gamma = \{S_i\}_{i=1,\dots,l}$, where l is the length of the sequence. When comparing dynamic shapes, it is often desirable to have a dissimilarity measure which is invariant to temporal misalignments and warping. This can be obtained for example via classical dynamic time warping, *e.g.* [20], obtaining a distance $d_{\mathrm{DS}}(\cdot,\cdot)$ between two dynamic shapes, computed using any given distance $d(\cdot,\cdot)$ between two static shapes.

6.1. Gait data

In this example we use the proposed approach for dynamic shape analysis of gait. The used dataset contains the same person performing different gait types: walking, jogging, running, running lifting the heel, and running lifting the knee. The data was obtained by filming a single person performing these activities with a fixed background. The silhouettes of the person in each frame is then extracted as a binary image of 65×45. The total database consist in 2380 segmented shapes.

Following [15], we consider a semicycle as a basic unit for processing. Let us denote by S_i the shape extracted from the i-th video frame, with $i = 1, \dots, N_F$, being N_F the number of frames. In order to define the boundaries of each semicycle, we choose a reference shape S_r, and then compute the sequence of distances d_i as the (L^2) distance from the i-th shape to the reference shape. Boundaries between adjacent semicycles are given by the local minima of this sequence. The selection of the reference frame is important, since it simplifies the search for the local minima. As in [15] we found that the best results are attained choosing the moment when the legs are aligned as the reference. There are 142 semicycles in the database.

In order to compare the distance between the semicycles, we follow the approach of [20]. Instead of aligning/warping each pair of whole dynamic shapes before comparing them, we align the semicycles against a reference one. First, this has a computational benefit, being N_{SC} the number of semicycles, we perform (dynamic time warping) DTW N_{SC} times, instead of N_{SC}^2 times. However our main reason for this partition will become apparent after the following discussion.

The Gaussian kernel can be computed just using a distance matrix among the semicycles. On the other hand, for applications that need to compute pre-images, the input space needs a vector space structure. Therefore, we want all our aligned semicycles to have the same length. This is easier to achieve by aligning all the sequences to a reference semicycle.

DTW performs the alignment repeating elements in the sequence that is evolving faster, in order to make it wait for the other sequence. This intuitive notion of speed is local, therefore elements can be added to both sequences. However if one of these sequences is overall slower, it is likely that DTW will not add any elements to it, and the final length of the aligned sequences will be that of the slowest one. This suggest to choose the longest semicycle as the reference one. Still we can not ensure that any elements will not be added to the longest sequence, increasing the final length. We overcome this problem removing the repetitions in the reference frame, which can be easily achieved in the DTW computation.

Following these preprocessing steps, we are ready to proceed with the dynamic shape analysis. The set of dynamic shapes, with the metric d_{DS} obtained from the DTW, constitutes a dynamic shape space. The dimensionality of this space is $l \times m_r \times m_c$, where l is the length of the aligned semicycles and m_r and m_c are the number of rows and columns of the signed distance functions used to represent the shapes. In our examples, $l = 19$, $m_r = 65$, $m_c = 45$. We use the Gaussian kernel in order to reduce dimensionality. The dimensionality of the mapped set depends on the decay of the eigenvalues of the kernel matrix K. The parameter σ plays an important role in this issue. Following [7, 17], we again compute σ as the average of the distances to the nearest neighbors.

This concludes the key processing to map the dynamic gaits to the feature Hilbert space. Applications such as clustering or classification can be done in this feature space. These kind of applications do not necessarily need the computation of pre-images, and might not even need the computation of the map itself following the kernel trick. We thereby center our attention in applications where we can demonstrate the use of the proposed pre-image framework, such as dynamic shape de-noising, reconstruction, and visualization. We select to exemplify the cases of DS averaging and reconstruction from occlusions.

Figure 2 shows the pre-images of the averages of feature space points corresponding to five types of gait (the actual pre-images are thresholded to yield a binary image). Each average was computed with 10 sequences. Note that the results correspond to recognizable gait sequences of each type.

The process of reconstructing an occluded dynamic sequence is as follows. First, the new sequence is mapped into the feature space using the Nyström extension. This is

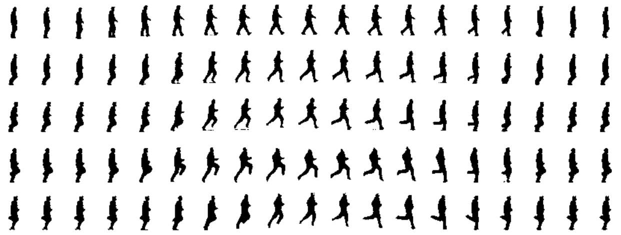


Figure 2. *Pre-images of the averages of 10 dynamic shapes of a person performing different types of gait computed in the feature space. Each row corresponds to a different activity: Walking, jogging, running, running lifting the knee, running lifting the heel.*

of course done after the training set was used to construct the map and the distance of the new dynamic shape to the training set is computed using the semicycles-based DTW as described above. Then, this mapped point is projected in the subspace spanned by its nearest neighbors. The pre-image of the projected point is then a reconstructed version of the original sequence. The results of this simple technique are displayed in Figure 3.

6.2. Lips data

We now use the dataset collected by Aharon & Kimmel, [1], to exemplify the use of the proposed pre-image method for upsampling of lips movement sequences. This dataset consists of sequences of images of the mouth of a single subject while pronouncing different syllables. Each syllable is a combination of a consonant with a vowel. There are 20 consonants and 6 vowels. Each sequence has approximately 30 frames. This yields a total number of more than 3400 frames. For more details about the segmentation and preprocessing of the images refer to [1].

In this example we consider each frame as a point in a high dimensional space. A syllable is then a curve in the manifold where these high dimensional points live.

To compare frames, we use the JBB metric, [1, 14], which is less sensitive to lighting changes. Using this metric between frames, we compute the kernel matrix and perform the (Gaussian) embedding. Once the data points are mapped into the feature space, their structure tends to represent the different positions of the lips, robustly to lighting changes.

For upsampling a new sequence, we first use the Nyström extension to map the available new samples onto the feature space, where we simply perform linear interpolation, and then compute the pre-image of the interpolated points with our proposed approach. To see the invariance to the strong lighting changes, we intentionally distort the samples that are going to be interpolated, simulating fast local brightness changes.

Figure 4 shows the result of upsampling a sequence by a factor of two. The sequence was downsampled and the even frames considered as missing frames, whereas the brightness-distorted odd frames were used to interpolate. It can be seen that linearly interpolating in the kernel space yields results without the brightness distortions that are present when working in the image space.

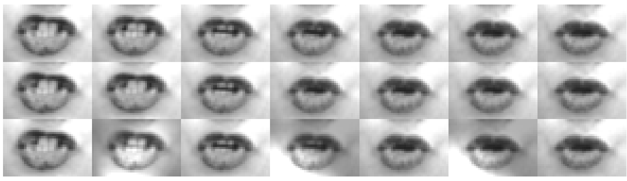


Figure 4. *Results of the interpolation for a test sequence. Top row, original sequence. Second row, even frames interpolated in the feature space. Third row, interpolation in the input space.*

7. Conclusions and future work

In this paper we presented a new approach for computing the pre-image of the mapping associated with kernel methods. Following the Nyström extension we showed the analogy between the out-of-sample and the pre-image tasks. This connection not only provided a new technique for pre-image computation, but also lead to new insights into the problem and new connections with prior techniques. As a consequence of this approach we derive a new way of estimating the kernel values between the unknown pre-image and the given training points. We compared the proposed frameworks with their corresponding counterparts in the existing literature, both numerically and perceptually. Depending on the parameters of the problem (width of the kernel, size and dimension of the training dataset), our results are comparable, if not better than those of the previous approaches. We presented simple uses of the method for dynamic shapes.

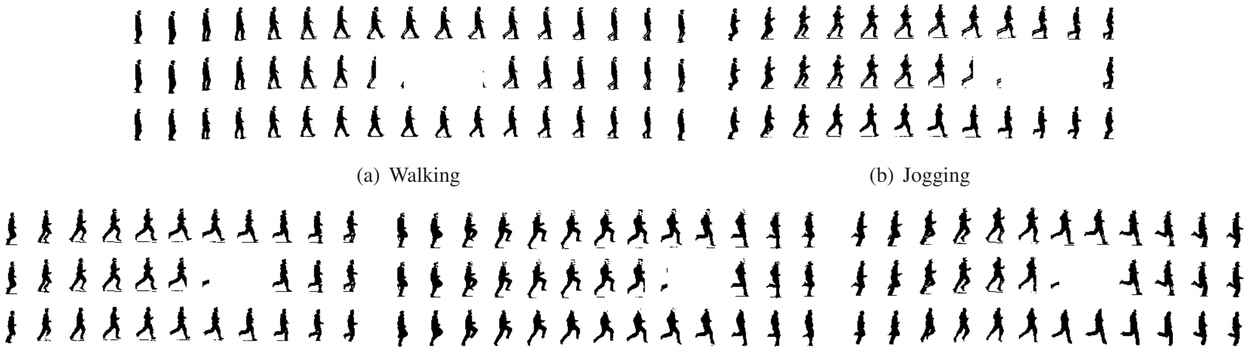
We are currently working on exploiting the proposed framework with different kernels and distance metrics between static and dynamic shapes. We are also pursuing theoretical estimates related to the accuracy of the Nyström based pre-image computation. This is an interesting issue related to the conditions under which the point cloud captures the intrinsic geometry of the manifold as well as with the ability of the mapping to learn it from the training set. Results in these directions will be reported elsewhere.

References

[1] M. Aharon and R. Kimmel. Representation analysis and synthesis of lip images using dimensionality reduction. *Int. J. Comput. Vision*, 67(3):297–312, 2006.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in NIPS 14*, 2002.

[3] Y. Bengio, O. Delalleau, N. Le Roux, J.-F. Paiement, P. Vincent, and M. Ouimet. Learning eigenfunctions links spec-



(a) Walking

(b) Jogging

(c) Running

(d) Lifting the knee

(e) Lifting the heel

Figure 3. *Reconstruction of occluded dynamic shapes. Results with 10 nearest neighbors. First row: original dynamic shape. Second row: occluded dynamic shape. Third row: reconstructed dynamic shape using the proposed pre-image framework.*

tral embedding and kernel PCA. *Neural Computation*, 16(10):2197–2219, 2004.

[4] Y. Bengio, J.-F. Paiement, and P. Vincent. Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps and Spectral Clustering. In *Advances in NIPS 16*, 2004.

[5] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman & Hall, London, 1994.

[6] D. Cremers. Dynamical statistical shape priors for level set based tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1262–1273, August 2006.

[7] D. Cremers, T. Kohlberger, and C. Schnörr. Nonlinear shape statistics via kernel spaces. *Lecture Notes in Computer Science*, 2191:269, 2001.

[8] S. Dambreville, Y. Rathi, and A. Tannenbaum. Shape-based approach to robust image segmentation using kernel PCA. In *Proc. of the IEEE Conference on CVPR*, 2006.

[9] S. Dambreville, Y. Rathi, and A. Tannenbaum. Statistical shape analysis using kernel PCA. In *IS&T/SPIE Symposium on Electronic Imaging*, 2006.

[10] D. Donoho and C. Grimes. Hessian eigenmaps: locally linear embedding techniques for high dimensional data. *PNAS*, 100(10):5591–5596, 2003.

[11] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Behavior classification by eigen-decomposition of periodic motions. *Pattern Recognition*, 38(7):1033–43, 2005.

[12] J. Gower. Adding a point to vector diagrams in multivariate analysis. *Biometrika*, 55(3):582–585, November 1968.

[13] J. H. Ham, D. D. Lee, S. Mika, and B. Schölkopf. A kernel view of the dimensionality reduction of manifolds. In *Proc. of the 21st ICML*, 2004.

[14] W. Jacobs, P. N. Belhumeur, and R. Basri. Comparing images under variable illumination. In *Proc. of the IEEE Conference on CVPR*, 1998.

[15] D. Kaziska and A. Srivastava. Cyclostationary processes on shape spaces for gait-based recognition. In *ECCV06*, pages 442–453, 2006.

[16] J. T. Kwok and I. Tsang. The pre-image problem in kernel methods. *IEEE Transactions on Neural Networks*, 15(6):1517–1525, November 2004.

[17] S. Lafon. *Diffusion Maps and Geometric Harmonics*. PhD thesis, Yale University, 2004.

[18] S. Lafon, Y. Keller, and R. R. Coifman. Data fusion and multicue data matching by diffusion maps. *IEEE Transactions on PAMI*, 28(11):1784–1797, November 2006.

[19] C. Lee and A. Elgammal. Gait tracking recognition using person-dependent dynamic shape model. In *7th Int.l Conf. Automatic Face and Gesture Recognition*, 2006.

[20] P. Maurel and G. Sapiro. Dynamic shape average. In *Proc. Second IEEE Workshop Variational, Geometric, and Level Set Methods in Computer Vision*, 2003.

[21] S. Mika, B. Schökopf, A. Smola, K. Müller, M. Scholz, and G. Rätsch. Kernel PCA and de-noising in feature spaces. In *Advances in Neural Information Processing Systems 11*, Cambridge, MA, USA, 1998. MIT Press.

[22] Y. Rathi, N. Vaswani, and A. Tannenbaum. A generic framework for tracking using particle filter with dynamic shape prior. *IEEE Transactions on IP*, 16(5):1370–82, 2007.

[23] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[24] B. Schölkopf and A. Smola. *Learning with Kernels. Support Vector Machines, Regularization, Optimization and Beyond.* MIT Press, Cambridge, MA, USA, 2002.

[25] B. Schölkopf, A. J. Smola, and K.-R. Müller. Kernel principal component analysis. *Advances in kernel methods: support vector learning*, pages 327–352, 1999.

[26] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[27] R. Urtasun, D. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *Conference on Computer Vision and Pattern Recognition*, June 2006.

[28] K. Q. Weinberger, F. Sha, and L. K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proc. of the 21st ICML*, pages 839–846, Banff, Canada, 2004.