Local Isomorphism to Solve the Pre-image Problem in Kernel Methods

Dong Huang[1,2], Yuandong Tian[1] and Fernando De la Torre[1]
[1]Robotics Institute, Carnegie Mellon University, USA
[2]University of Electronic Science and Technology of China, China
dghuang@andrew.cmu.edu, yuandong@andrew.cmu.edu, ftorre@cs.cmu.edu,

Abstract

Kernel methods have been popular over the last decade to solve many computer vision, statistics and machine learning problems. An important, both theoretically and practically, open problem in kernel methods is the pre-image problem*. The pre-image problem consists of finding a vector in the input space whose mapping is known in the feature space induced by a kernel. To solve the pre-image problem, this paper proposes a framework that computes an isomorphism between local Gram matrices in the input and feature space. Unlike existing methods that rely on analytic properties of kernels, our framework derives closed-form solutions to the pre-image problem in the case of non-differentiable and application-specific kernels. Experiments on the pre-image problem for visualizing cluster centers computed by kernel k-means and denoising high-dimensional images show that our algorithm outperforms state-of-the-art methods.*

1. Introduction

In recent years, there has been a lot of interest in the study of kernel methods [1,5,19,20] in the computer vision, statistics and machine learning communities. In particular, kernel methods have proven to be useful in many computer vision problems [14] such as object classification, action recognition, image segmentation and content based image retrieval. In kernel methods, a non-linear mapping $\varphi(\cdot)$ is used to transform the data $\mathbf{X}$ in the input space to a feature space where linear methods can be applied. Many standard linear algorithms such as Principal Component Analysis (PCA) [12], Linear Discriminant Analysis (LDA) [8] and Canonical Component Analysis (CCA) [11] can be extended to model the non-linear structure in the data without local minima using kernel methods.

In kernel methods, the mapping is typically never computed explicitly but implicitly with a kernel function, $k(\mathbf{x}_1, \mathbf{x}_2) = \varphi(\mathbf{x}_1)^T \varphi(\mathbf{x}_2)$ as the inner product in the feature space. By the Representer Theorem, every symmetric positive definite function defines an inner product in some Hilbert feature space which can be implicitly mapped from the input space. An important yet nontrivial problem in kernel methods called the pre-image problem is to find the inverse mapping φ^{-1} from the feature space to the input space. Finding a closed-form solution to the pre-image problem is both theoretically interesting and useful in many applications, such as feature space visualization and image denoising. Several challenges include: **(a)** the exact pre-image does not always exist and it might not be unique, and an approximation needs to be made; **(b)** there is no closed-form and smooth solution for complicated and application-specific kernels. **(c)** The pre-image of a test sample is usually biased towards the training data and loses the test-specific features.

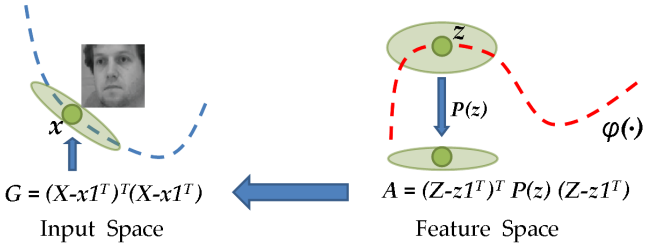

Figure 1. The local isomorphism between the Gram matrices from the feature space to the input space. Our solution to the pre-image and denoising problem is based on this connection. Specifically, the pre-image $\mathbf{x}$ of a feature vector $\mathbf{z} = \varphi(\mathbf{x})$ can be obtained by firstly computing the local Gram matrix $\mathbf{A}$ at $\mathbf{z}$ using training samples, and then finding the pre-image $\mathbf{x}$ so that its own local Gram matrix $\mathbf{G}$ is matched with that of $\mathbf{z}$.

This paper addresses the pre-image problem by building a *local isomorphism* between the input and feature space using local Gram matrices (Fig. 1). The Gram matrices are respectively computed in both spaces using nearby data points, modeling important local structural information, i.e. linear or non-linear correlations between nearby samples.

By introducing a local metric $\mathbf{P}(\mathbf{z})$ in the feature space, the two local Gram matrices above can be matched. This local matching implicitly builds a bidirectional relation between both spaces, making it possible to solve the pre-image problem. Specifically, the pre-image $\mathbf{x}$ of a feature vector $\mathbf{z} = \varphi(\mathbf{x})$ can be obtained by firstly computing the local Gram matrix at $\mathbf{z}$ using training samples, and then finding the pre-image $\mathbf{x}$ so that its own local Gram matrix is matched with that of $\mathbf{z}$. In addition, we can also use this structural relationship for image denoising. Image denoising can be achieved by matching the local Gram matrices of a test pair $\{\mathbf{x}, \mathbf{z}\}$.

There are three advantages of our work: **(1)** The pre-image problem can be solved in closed-form; **(2)** any feature mapping $\varphi(\cdot)$ can be modeled regardless of whether its kernel function has closed-form and/or is differentiable, substantially broadening the range of its usage compared to [13, 16]; **(3)** The test-specific denoising preserves the subtle visual characteristics of the test images.

The rest of the paper is organized as follows: Section 2 reviews previous works on the pre-image problem. Section 3 introduces the main formulation of our work, and Section 4 describes the applications of our method to solve the pre-image and denoising problems. Section 5 shows experimental results and Section 6 concludes the paper.

2. Previous Work

This section reviews previous methods to solve the pre-image problem according to their optimization criteria.

The first set of algorithms that solve the pre-image problem minimize the distance between the image of the pre-image and the test data point in feature space. Mika et al. [16] found an approximate pre-image for a Gaussian kernel using a fixed-point iteration. This method is sensitive to initialization and susceptible to local minima. Rathi et al. [18] added a preprocessing step [16] by projecting the test sample onto the subspace of the training set. Both methods above assume the kernel function is normalized, differentiable and with explicit derivatives. Kowk and Tsang [13] applied multidimensional scaling (MDS) and reconstructed the pre-image using the local tangent space of the training samples. This method requires a closed-form relationship between the distances in the feature space and the Euclidean distances in the input space.

The second set of algorithms regularize the feature space distance with some prior information. Nguyen and De la Torre [17] proposed a robust kernel PCA method that handles missing data and intra-sample outliers. They derived an error function where the image of the pre-image is close to both the image of the noisy data point and its own projection in the kernel principal subspace. Zheng et al. [21] computed the pre-image regularized by a weakly supervised prior that weights positive training samples over negative ones.

The third set of algorithms directly model the inverse transformation from the feature space to the input space. Honeine and Richard [10] proposed a direct method to learn a global linear transformation between the input space and the feature space. Thus, the pre-image of the test point can be solved analytically. Bakir et al. [3] proposed to learn a global transformation for the pre-image in a regression fashion. However, learning global parameters inevitably results in biased pre-images.

Other extensions include Arias et al. [2], which connects the pre-image problem with the out-of-sample extension using the Nyström method [4]. A new feature space vector for the test data point is computed after normalizing the training data in the feature space onto the unit sphere. Then the pre-image can be computed by [16] or [13] using the new feature space vector.

3. The Local Gram Matrix Isomorphism

This section describes the isomorphism between the input and the feature space using the local Gram matrix.

3.1. Notation

Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \Re^{d \times n}$ (see the footnote for notations[1]) be a matrix containing n d-dimensional input data points. $\mathbf{K} \in \Re^{n \times n}$ denotes the kernel matrix such that each element $k_{ij} = \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j)$ $(i, j = 1, \cdots, n)$ measures the similarity between two points using a kernel function. $\varphi(\cdot)$ is typically a nonlinear function that transforms $\mathbf{X}$ into a (usually) higher dimensional feature space.

3.2. The Gram Matrix

Given a set of data points in the input space, the weighted Gram matrix $\mathbf{G}(\mathbf{x}; \mathbf{P})$ centered at some point $\mathbf{x}$ is defined as follows:

$$\mathbf{G}(\mathbf{x}; \mathbf{P}) = (\mathbf{X} - \mathbf{x}\mathbf{1}^T)^T \mathbf{P} (\mathbf{X} - \mathbf{x}\mathbf{1}^T). \quad (1)$$

Each element $g_{ij}(\mathbf{x}; \mathbf{P}) = (\mathbf{x}_i - \mathbf{x})^T \mathbf{P} (\mathbf{x}_j - \mathbf{x})$ of $\mathbf{G}$ represents the inner product between the data sample $\mathbf{x}_i$ and $\mathbf{x}_j$ centered at $\mathbf{x}$, weighted by a positive semi-definite matrix $\mathbf{P}$ which is dependent on particular choice of $\mathbf{x}$. This Gram matrix $\mathbf{G}(\mathbf{x}; \mathbf{P})$ represents the first and second-order information of the data distribution centered at $\mathbf{x}$ in the input space. By only selecting neighbors of $\mathbf{x}$ in the training set $\mathbf{X}$, we can define a *local* Gram matrix that encodes the local structure of the data at $\mathbf{x}$.

[1] Bold capital letters denote matrices $\mathbf{X}$, bold lower-case letters a column vector $\mathbf{x}$. $\mathbf{x}_j$ represents the j^{th} column of the matrix $\mathbf{X}$. All non-bold letters represent scalar variables. x_{ij} denotes the scalar in the row i and column j of the matrix $\mathbf{X}$ and the scalar i^{th} element of a column vector $\mathbf{x}_j$. $\|\mathbf{x}\|_2$ denotes the norm of the vector $\mathbf{x}$. $\mathrm{tr}(\mathbf{A}) = \sum_i a_{ii}$ is the trace of $\mathbf{A}$, $\det(\mathbf{A})$ is the determinant of A and $\mathrm{diag}(\mathbf{x})$ denotes a operator that generates a diagonal matrix with the elements of the vector $\mathbf{x}$. $\|\mathbf{A}\|_F^2 = \mathrm{tr}(\mathbf{A}^T\mathbf{A}) = \mathrm{tr}(\mathbf{A}\mathbf{A}^T)$ designates the Frobenius norm of matrix $\mathbf{A}$. $\mathbf{1}$ is a vector with all elements 1.

Intuitively, if a smooth surface is represented by discrete training samples and the local Gram matrix is computed from a point on the surface, then $\mathbf{G}(\mathbf{x};\mathbf{P})$ represents not only the tangent directions on the surface but also the curvature at $\mathbf{x}$. In fact, one can regard $\mathbf{G}(\mathbf{x};\mathbf{P})$ as an *empirical* Hessian matrix computed at $\mathbf{x}$ from the training samples.

Similarly, we can apply the Gram matrix definition from Eqn. 1 to the feature space where the local metric $\mathbf{P}$ is dependent on centering the point in the feature space. Observe that the possible infinite dimensional feature space is sampled by a finite number of training data. To smooth the data in the feature space, we apply the Nyström method [4] to obtain a low-dimension representation $\mathbf{Y}$ of the feature space, by eigen-decomposing the kernel matrix $\mathbf{K} = \mathbf{V}\Lambda\mathbf{V}^T$, where $\Lambda \in \Re^{m\times m}$ is the diagonal matrix of m largest eigenvalues, and the columns of $\mathbf{V} \in \Re^{n\times m}$ are the m eigenvectors ($m \equiv rank(\mathbf{K}) \leq n$). Then the feature data $\varphi(\mathbf{X})$ can be represented by

$$\mathbf{Y} = \Lambda^{-1/2}\mathbf{V}^T\mathbf{K}, \tag{2}$$

where $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_n] \in \Re^{m\times n}$ contains m-dimensional representations of the data points in the feature space such that $\mathbf{K} = \mathbf{Y}^T\mathbf{Y}$, i.e., the inner product is preserved. Thus the definition of the local Gram matrix (Eqn. 1) can be applied in this low-dimensional representation of the feature space.

3.3. The Criterion for Establishing Isomorphism

Given a test sample $\mathbf{x}_t$ in the input space, its kernel image $\varphi(\mathbf{x}_t)$ can be represented as

$$\mathbf{y}_t = \Lambda^{-1/2}\mathbf{V}^T\mathbf{k}(\cdot, \mathbf{x}_t), \tag{3}$$

in the low-dimensional space, where $\mathbf{k}(\cdot, \mathbf{x}_t) = [k(\mathbf{x}_1, \mathbf{x}_t), k(\mathbf{x}_2, \mathbf{x}_t), \ldots, k(\mathbf{x}_n, \mathbf{x}_t)]^T \in \Re^{n\times 1}$ contains the inner-products in the feature space between the test sample $\mathbf{x}_t$ and the training set $\mathbf{X}$. In this work, we aim to match the input space Gram matrices $\mathbf{G}_t \equiv \mathbf{G}(\mathbf{x}_t;\mathbf{I})$ and the feature space Gram matrix $\mathbf{A}_t(\mathbf{P}_t) \equiv \mathbf{G}(\mathbf{y}_t;\mathbf{P}_t)$ by a proper choice of a positive-definite *local metric* $\mathbf{P}_t \in \Re^{m\times m}$. The matrix $\mathbf{P}_t$ essentially parameterizes the local isomorphism between the two spaces with different ambient dimensions (but the same intrinsic dimension), as shown geometrically in Fig. 1.

We emphasize that this isomorphism leads to a locally-defined connection between the two spaces, which is used in the rest of the paper. The local neighborhood of a data point is defined by the data points around it, and it can capture complex nonlinear structure. Our main assumption is that the neighborhood structure defined in the feature space is similar to the one in the input space. Observe that the kernel-induced feature mapping connecting the two spaces is continuous and preserves the (topological) neighborhood structure. Moreover, the data point in the input space is often unknown and the associated neighborhood structure has to be inherited from the feature space.

Note that alternatively, it is also possible to build the mapping reversely from the input space to the feature space by matching $\mathbf{G}(\mathbf{x}_t;\mathbf{P}_t)$ and $\mathbf{G}(\varphi(\mathbf{x}_t);\mathbf{I})$, which seems to be better since the Nyström method is no longer needed. However, the dimension of the input space is typically high(e.g., several ten thousands for raw image pixels). As a result, $\mathbf{P}_t$ contains many free parameters and typically there is very little training data to constrain them.

Following previous work on the Gaussian Processes Latent Variable Model [15], the matching between two Gram matrices $\mathbf{G}_t$ and $\mathbf{A}_t$, is defined by the solution that maximizes the following criterion parameterized by $\mathbf{P}_t$:

$$J(\mathbf{P}_t;\mathbf{X},\mathbf{Y},\mathbf{x}_t,\mathbf{y}_t) = \frac{(2\pi)^{n/2}\exp\left(-\frac{1}{2}\mathrm{tr}\left[\mathbf{A}_t^{-1}\mathbf{G}_t\right]\right)}{\det(\mathbf{A}_t)^{1/2}}. \tag{4}$$

where

$$\mathbf{A}_t = \mathbf{G}(\mathbf{y}_t;\mathbf{P}_t) = (\mathbf{Y}^{(t)} - \mathbf{y}_t\mathbf{1}^T)^T\mathbf{P}_t(\mathbf{Y}^{(t)} - \mathbf{y}_t\mathbf{1}^T)$$

is a function of $\mathbf{P}_t$, and

$$\mathbf{G}_t = \mathbf{G}(\mathbf{x}_t;\mathbf{I}) = (\mathbf{X}^{(t)} - \mathbf{x}_t\mathbf{1}^T)^T(\mathbf{X}^{(t)} - \mathbf{x}_t\mathbf{1}^T),$$

where $\mathbf{Y}^{(t)}$ contains the nearest neighbors of $\mathbf{y}_t$ in the feature space and $\mathbf{X}^{(t)}$ is the subset in $\mathbf{X}$ corresponding to $\mathbf{Y}^{(t)}$. Observe that this is a measure of normalized correlation between two covariances. In practice, when the Gram matrix $\mathbf{A}_t$ in the feature space is rank deficient, we can add a regularization term as $\mathbf{A}_t \leftarrow \mathbf{A}_t + \beta\mathbf{I}$ (where $\beta > 0$).

Eqn. (4) serves as the key component of our method. Multiple tasks are successfully unified using Eqn. (4). For instance, we can build the local connection between two spaces by optimizing $\mathbf{P}_t$, analytically solve the pre-image problem by optimizing $\mathbf{x}_t$ in $\mathbf{G}_t$ given the image $\mathbf{y}_t$ and $\mathbf{P}_t$; we can also perform data denoising by alternatively optimizing the denoised version of $\mathbf{x}_t$ and $\mathbf{y}_t$. Note originally Eqn. (4) comes from Gaussian Processes Latent Variable Model [15], where the variance of a set of latent variables in the low-dimensional space to be learned fit with the variance of the observation. However, we use it here for a different purpose: to calibrate the Gram matrices in two spaces.

Computing the partial derivative of $\log[J(\mathbf{P}_t;\mathbf{X},\mathbf{Y},\mathbf{x}_t,\mathbf{y}_t)]$ with respect to $\mathbf{P}_t$, we obtain:

$$\begin{aligned}&\frac{\partial \log[J(\mathbf{P}_t;\mathbf{X},\mathbf{Y},\mathbf{x}_t,\mathbf{y}_t)]}{\partial \mathbf{P}_t}\\ =\ &\mathbf{A}_t^{-1}\mathbf{G}_t\mathbf{A}_t^{-1}(\mathbf{Y}^{(t)} - \mathbf{y}_t\mathbf{1}^T)^T(\mathbf{Y}^{(t)} - \mathbf{y}_t\mathbf{1}^T)\\ &-\mathbf{A}_t^{-1}(\mathbf{Y}^{(t)} - \mathbf{y}_t\mathbf{1}^T)^T(\mathbf{Y}^{(t)} - \mathbf{y}_t\mathbf{1}^T).\end{aligned} \tag{5}$$

where $\mathbf{P}_t$ has a closed-form solution:

$$\begin{aligned}\mathbf{P}_t\ =\ &\left[(\mathbf{Y}^{(t)} - \mathbf{y}_t\mathbf{1}^T)^T\right]^\dagger(\mathbf{X}^{(t)} - \mathbf{x}_t\mathbf{1}^T)^T\\ &(\mathbf{X}^{(t)} - \mathbf{x}_t\mathbf{1}^T)(\mathbf{Y}^{(t)} - \mathbf{y}_t\mathbf{1}^T)^\dagger,\end{aligned} \tag{6}$$

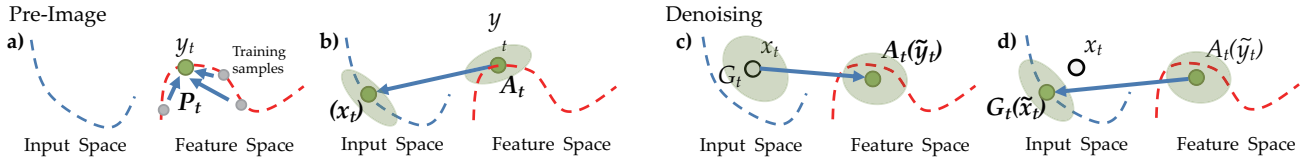

Figure 2. The workflow of pre-image and denoising in the framework of local Gram matrix isomorphism. **Pre-image**(left column): **(a)** Given the feature vector $\mathbf{y}_t$, firstly the local metric $\mathbf{P}_t$ is estimated from its neighboring training samples (Eqn. (7)); **(b)** then the feature space Gram matrix $\mathbf{A}_t$ is matched with the input space Gram matrix $\mathbf{G}_t = \mathbf{G}_t(\mathbf{x}_t)$ and the optimal $\mathbf{x}_t$, as the pre-image of $\mathbf{y}_t$, is obtained (Eqn. (8)). **Denoising**(right column): **(c)** Given a noisy vector $\mathbf{x}_t$ in the input space, its Gram matrix $\mathbf{G}_t$ is matched with the Gram matrix $\mathbf{A}_t = \mathbf{A}_t(\widetilde{\mathbf{y}}_t)$ in the feature space (Eqn. (9)), where $\widetilde{\mathbf{y}}_t$ is expected to be a denoised version of the image of $\mathbf{x}_t$; **(d)** The Gram matrix $\mathbf{G}_t(\widetilde{\mathbf{x}}_t)$ is again matched with $\mathbf{A}_t(\widetilde{\mathbf{y}}_t)$ by optimizing $\widetilde{\mathbf{x}}_t$ (Eqn. (11)), which is the final denoised version of $\mathbf{x}_t$.

where $\mathbf{M}^\dagger$ is the pseudo-inverse of a matrix $\mathbf{M}$.

4. Applications

This section describes two applications, closed-form pre-image and image denoising, that make use the local isomorphism defined in Eqn. (4).

4.1. Local Gram Preserving Pre-image

Given $\mathbf{y}_t$ in the feature space, finding its pre-image $\widetilde{\mathbf{x}}_t$ using Eqn. (4) requires the knowledge of the local metric $\mathbf{P}_t$. A joint optimization over both $\widetilde{\mathbf{x}}_t$ and $\mathbf{P}_t$ is not feasible because for any $\widetilde{\mathbf{x}}_t$ in the input space there would always be one $\mathbf{P}_t$ that matches the local structure near $\mathbf{y}_t$ in the feature space. Instead, we first estimate $\mathbf{P}_t$ and then solve $\widetilde{\mathbf{x}}_t$, as shown in Fig. 2.

Consider $\mathcal{N}_t$ be the subset containing the neighbors of $\mathbf{y}_t$ in $\mathbf{Y}$. We assume that the local metric changes smoothly due to the continuity of local metric structure and compute the local metric at $\mathbf{y}_t$ as a weighted combination of the neighboring metrics, that is:

$$\mathbf{P}_t = \frac{1}{\sum_{i=1}^{|\mathcal{N}_t|} \alpha_i} \sum_{i=1}^{|\mathcal{N}_t|} \alpha_i \mathbf{P}_i, \tag{7}$$

where the local metric $\mathbf{P}_i$ for a particular neighboring training sample $\mathbf{y}_i \in \mathcal{N}_t$ is computed using Eqn. (6) and the weight coefficient is typically set as $\alpha_i = exp\{-(\mathbf{y}_t - \mathbf{y}_i)^T \mathbf{P}_i (\mathbf{y}_t - \mathbf{y}_i)/\delta^2\}$, with δ controlling the smoothness in the neighborhood. Then given $\mathbf{P}_t$, Eqn. (4) can be optimized with respect to $\mathbf{x}_t$ in $\mathbf{G}_t$, and the solution can be found analytically as:

$$\widetilde{\mathbf{x}}_t = \frac{\mathbf{X}^{(t)} \mathbf{A}_t^{-1} \mathbf{1}}{\mathbf{1}^T \mathbf{A}_t^{-1} \mathbf{1}}, \tag{8}$$

where $\mathbf{A}_t = (\mathbf{Y}^{(t)} - \mathbf{y}_t \mathbf{1}^T)^T \mathbf{P}_t (\mathbf{Y}^{(t)} - \mathbf{y}_t \mathbf{1}^T)$. The complexity for solving Eqn. (8) is fairly low considering that we only used neighboring data points ($|\mathcal{N}_t| \ll n$). We emphasize that our proposed approach is purely data-driven and does not put any special requirements on the kernel function, such as being invertible and differentiable as in previous works (e.g. [16] [13] [17]).

4.2. Joint De-noising in the Input and Feature Space

Using Eqn. (4) we can also solve the denoising problem by jointly working in the input and feature space. Given a noisy input space vector $\mathbf{x}_t$, its feature space representation $\mathbf{y}_t$ will inherent the noise from the input space. Therefore, $\mathbf{y}_t$ should be denoised before estimating the noise free pre-image $\widetilde{\mathbf{x}}_t$. In this case, we formulate a two-step process for joint denoising. In the first step we obtain a denoised feature vector $\widetilde{\mathbf{y}}_t$ (note this is different from the direct kernel mapping) from $\mathbf{y}_t$. In the second step we obtain the final denoised input space vector $\widetilde{\mathbf{x}}_t$ from $\widetilde{\mathbf{y}}_t$, as shown in Fig. 2.

However, denoising this way typically leads to the over-smoothing problem, i.e., the denoising algorithm not only removes the noise but also eliminates the specific characteristics of the test sample, especially when such characteristics are not present in the training set (e.g., pimples or glasses on a face). Essentially, this problem is due to the lack of training samples and is ill-posed. Since the test-specific information not present in the training set cannot be modeled, it is difficult to factorize this information from the noise. But practically, a regularization term can be added to keep a trade-off between denoising and preserving test-specific characteristics. Our idea of using the trade-off parameter follows previous methods such as [17].

Specifically, given a noisy test sample $\mathbf{x}_t$ in the input space, we first compute its Gram matrix $\mathbf{G}_t$ and its image $\mathbf{y}_t$, then obtain $\mathbf{P}_t$ using Eqn. (6), and finally optimize the following objective (Eqn. (9)) with respect to $\widetilde{\mathbf{y}}_t$, the denoised feature vector. In fact, the objective for denoising the feature space is Eqn. (4) plus a regularization that combines $\widetilde{\mathbf{y}}_t$ with the noisy $\mathbf{y}_t$:

$$\max_{\widetilde{\mathbf{y}}_t} E^F(\widetilde{\mathbf{y}}_t) = \frac{\exp\left\{-\frac{1}{2}\mathrm{tr}\left[\left(\widetilde{\mathbf{A}}_t + \lambda \mathbf{R}_t^F\right)^{-1} \mathbf{G}_t\right]\right\}}{(2\pi)^{n/2} \det\left(\widetilde{\mathbf{A}}_t + \lambda \mathbf{R}_t^F\right)^{1/2}} \tag{9}$$

where $\lambda \in [0, 1]$ is the regularization parameter, $\widetilde{\mathbf{A}}_t = (\mathbf{Y}^{(t)} - \mathbf{y}_t \mathbf{1}^T)^T \mathbf{P}_t (\mathbf{Y}^{(t)} - \mathbf{y}_t \mathbf{1}^T)$ and $\mathbf{R}_t^F = \mathbf{1}(\widetilde{\mathbf{y}}_t - \mathbf{y}_t)^T \mathbf{P}_t (\widetilde{\mathbf{y}}_t - \mathbf{y}_t) \mathbf{1}^T$. The regularization matrix $\mathbf{R}_t^F$ is necessary in practice to avoid rank-deficiency of $\mathbf{A}_t$.

Computing the derivative of E^F with respect to $\widetilde{\mathbf{y}}_t$ and setting it to zero, we obtain the closed-form solution of $\widetilde{\mathbf{y}}_t$:

$$\widetilde{\mathbf{y}}_t = \frac{(\mathbf{Y}^{(t)}\mathbf{G}_t^{-1}\mathbf{1} + \lambda \mathbf{y}_t \mathbf{1}^T \mathbf{G}_t^{-1}\mathbf{1})}{(1+\lambda)\mathbf{1}^T\mathbf{G}_t^{-1}\mathbf{1}}. \tag{10}$$

Similarly, in the second step, we estimate the pre-image $\widetilde{\mathbf{x}}_t$ of $\widetilde{\mathbf{y}}_t$ in the input space by the following optimization:

$$\max_{\widetilde{\mathbf{x}}_t} E^I(\widetilde{\mathbf{x}}_t) \quad = \quad \frac{\exp\left\{-\frac{1}{2}\mathrm{tr}\left[\widetilde{\mathbf{A}}_t^{-1}\left(\widetilde{\mathbf{G}}_t + \lambda \mathbf{R}_t^I\right)\right]\right\}}{(2\pi)^{n/2}\det\left(\widetilde{\mathbf{A}}_t\right)^{1/2}},$$

where $\widetilde{\mathbf{G}}_t = (\mathbf{X}^{(t)} - \widetilde{x}_t\mathbf{1}^T)^T(\mathbf{X}^{(t)} - \widetilde{x}_t\mathbf{1}^T)$ and $\mathbf{R}_t^I = (\widetilde{\mathbf{x}}_t - \mathbf{x}_t)^T(\widetilde{\mathbf{x}}_t - \mathbf{x}_t)\mathbf{1}\mathbf{1}^T$. Computing the derivative of E^I with respect to $\widetilde{\mathbf{x}}_t$ and set it to zero, we have the final denoising result of $\mathbf{x}_t$:

$$\widetilde{\mathbf{x}}_t = \frac{\mathbf{X}^{(t)}\widetilde{\mathbf{A}}_t^{-1}\mathbf{1} + \lambda \mathbf{x}_t \mathbf{1}^T \widetilde{\mathbf{A}}_t^{-1}\mathbf{1}}{(1+\lambda)\mathbf{1}^T\widetilde{\mathbf{A}}_t^{-1}\mathbf{1}}. \tag{11}$$

5. Experimental Results

This section provides qualitative (visual) and quantitative comparisons between our method and previous works in two problems: finding pre-images of cluster centers after clustering with kernel k-means using non-differentiable kernels and images denoising, both using the CMU Multi-PIE database [9].

Figure 3. Example of normalized face images under different poses and illuminations.

5.1. Pre-image for Visualizing Cluster Centers

This section describes how to compute the pre-image of cluster centers after using kernel k-means [7] for clustering. In this case, we used two types of kernels: differentiable and not differentiable. The aim of this experiment is to show our method can visualize the cluster centers using a non-differentiable kernel, while it is unclear how other methods can do it.

We selected a subset of 260 images belonging to the Session 1 in the CMU MultiPIE database. This subset contains images taken from 13 view angles, each with 20 illumination conditions. Each image was cropped around the face areas and normalized to the size of 64×64. Fig. 3 shows some examples of the normalized images. With kernel k-means, the 260 images were clustered in different poses, and our task is to visualize the cluster centers. Ideally, if we select illumination-invariant measurement of similarity between images, each cluster should correspond to one view angle.

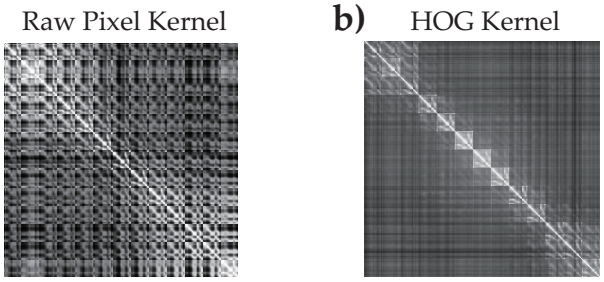

Figure 4. Visualization of kernel matrices using **(a)** the raw pixel distance and **(b)** the HOG distance. The samples are grouped according to the facial poses. Note that the HOG kernel shows a blocky structure, and thus, is better for clustering using kernel k-means and spectral relaxations. However, its kernel function is not differentiable.

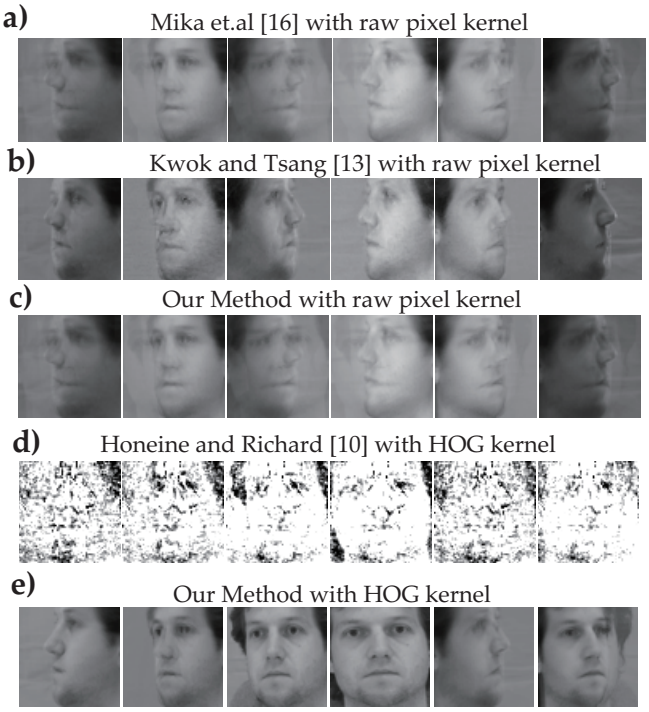

Figure 5. Visual comparison of clustering centers obtained by different methods. The first three rows are the pre-images of the cluster centers obtained by raw pixel kernel using **(a)** Mika et al. [16], **(b)** Kwok and Tsang [13] and **(c)** our method respectively. The last two rows show the pre-image by **(d)** Honeine and Richard [10] and **(e)** our method using the HOG kernel, which is the only valid visualization the pose cluster centers.

In this application two different kernels were used. One used the RBF function on the raw pixels (differentiable) and the other on the Histogram-Of-Gradient (HOG) (non-

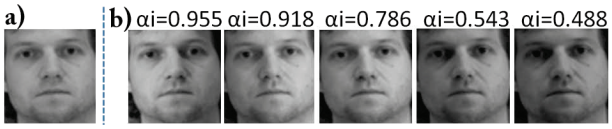

Figure 6. Pre-images of the 4^{th} cluster center in Fig. 5(e) computed using **(a)** $\mathbf{P}_t$ (Eqn. (7)) and using **(b)** several $\mathbf{P}_i$s associated with $\mathbf{y}_i$s (the neighbors of $\mathbf{y}_t$ in the feature space). The weight coefficient α_i on top of each image in **(b)** measures the similarity between $\mathbf{y}_t$ and $\mathbf{y}_i$.(see Eqn. (7)).

differentiable) [6]:

$$k^{\mathrm{RAW}}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma^2}\right\}, \quad (12)$$

$$k^{\mathrm{HOG}}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\frac{\|\mathbf{h}(\mathbf{x}_i) - \mathbf{h}(\mathbf{x}_j)\|_2^2}{\sigma^2}\right\}, \quad (13)$$

where $\mathbf{h}(\cdot)$ is the function computing HOG. Typically, the Euclidean distance between raw images may be unreliable and sensitive to illumination conditions, while the HOG distance is illumination-invariant (see Fig. 4). We selected the bandwidth σ for both kernels as the mean of the pairwise Euclidean distances.

Fig. 5 (a)-(c) visualize the pre-images of the cluster centers obtained using a raw pixel kernel (Eqn. (12)) [16], [13] and our method, (d)-(e) show the pre-images by [10] and our method using the HOG kernel. The neighbor number in both [13] and our method are selected as the number of data points within the mean pairwise Euclidean distances of the training set.

We can see that using the differentiable raw pixel kernel, kernel k-means provides poor clustering results, as indicated by the first three rows in Fig. 5 (a)-(c). Note, there is no point to compare the sharpness of these images. They are all supposed to be blurry because each cluster contains face images with various poses and illuminations. On the other hand, the HOG kernel provides a similarity that is more robust to illumination. However, the cluster centers cannot be visualized with existing methods because the kernel function is non-differentiable. [10] also failed to solve the pre-image problem because it computes the pre-image with global linear transformation, see Fig. 5 (d). Unlike existing work, our method can handle this important case, see Fig. 5 (e).

A key assumption of Eqn. (7) is the local smoothness of the metric $\mathbf{P}_t$. To verify this, in Fig. 6, we reconstructed $\mathbf{x}_t$ using **(a)** its associated $\mathbf{P}_t$ and **(b)** using $\mathbf{P}_i$, i.e. the metric of neighboring points y_i of y_t. As shown in Fig. 5, the reconstructed images are similar, especially when using $\mathbf{P}_i$ computed from a close neighbor $\mathbf{y}_i$ (characterized by higher $\boldsymbol{\alpha}_i$ in Eqn. (7)). The similarity in image appearances demonstrates $\mathbf{P}_t$ is locally smooth over the feature space.

5.2. Test-data-specific Denoising

This experiment compares the performance of our algorithm with Mika et al.'s fixed point method [16], Kwok and Tsang [13], Robust KPCA by Nguyen and De la Torre [17] and the direct global linear method by Honeine and Richard [10] on the image denoising problem. We show that our method removes the noise of the test image while preserving information of the original image, and provides better quantitative and visual results.

We selected a subset of frontal faces with the frontal illumination from the CMU MultiPIE database [9] containing 249 neutral faces, 249 smiling faces from Session 1, 203 surprise faces from Session 2, 203 squint faces from Session 2, 228 disgust faces from Session 3 and 239 scream faces from Session 4 respectively. All selected faces have been manually labeled with 66 points and warped towards a standard face. In this experiment, both the iconic variations (e.g. types of glasses, beards and eyebrows) and the expression variations (e.g. wrinkles on the cheek) are considered as the image-specific characteristics that are interesting to preserve.

We used 50% of the faces for training (i.e. 125 for neutral face, 125 for smile, 102 for surprise, 102 for squint, 114 for disgust and 120 for scream) and the remaining 50% for testing. All test data points were corrupted by Gaussian additive noise with standard deviation of 0.06. For the noisy and all the denoised images, two measures were used as performance measure: Average Pixel Error (APE) and the Signal to Noise Ratio (SNR):

$$APE = \frac{\|\widetilde{\mathbf{x}}_t - \mathbf{x}_t^0\|_1}{d}, \qquad SNR = \frac{\|\widetilde{\mathbf{x}}_t - \mathbf{x}_t\|_2^2}{\|\widetilde{\mathbf{x}}_t - \mathbf{x}_t^0\|_2^2} \quad (14)$$

where d is the number of pixels and $\mathbf{x}_t^0$ is the original clean test image. SNR defined in Eqn. (14) measures the ability to push the denoised image towards the clean image from the noisy image. Ideally, a good denoised image should have low APE, high SNR and be photo-realistic.

All the compared methods used the Gaussian kernel with the bandwidth parameter selected as the mean pairwise Euclidean distances of the training data. The number neighbor samples in both [13] and our method is selected as the number of data points within the mean pairwise Euclidean distances of the training set. All the methods used Kernel PCA to model the subspace of image variation, and for denoising, it kept all the components in the feature space. Both [17] and our method have a trade-off parameter that balances noise reduction with test-specific characteristics. We decide to show the results at their highest SNRs respectively, i.e. $c = 3$ for [17] and $\lambda = 0.1$ for our method.

Examples of the original clean image, noisy image, and denoised results of all the compared algorithms are shown in Fig. 7. Because Mika et al.'s [16] and Kwok and Tsang's

Figure 7. Examples of denoising face images. Columns from left to right: **(1)** the original test image, **(2)** image corrupted by Gaussian noise, **(3)** the result of Mika et al. [16], **(4)** Kwok&Tsang [13], **(5)** Nguyen&De la Torre [17], **(6)** Honeine&Richard [10] and **(7)** our method. In each column, the first number in brackets is the Average Pixel Error (APE) and the second is Signal to Noise Ratio (SNR).

Table 1. Denoising results on Multi-PIE database measured by Average Pixel Error (APE) and Signal to Noise Ratio (SNR).

	Noisy	Mika et al. [16]	Kwok& Tsang [13]	Nguyen& De la Torre [17]	Honeine& Richard [10]	Our Method
Neutral	APE:12.14 SNR:0	9.28 ± 2.77 3.06	10.60 ± 2.84 2.57	8.53 ± 2.58 2.30	26.90 ± 11.83 1.18	$\mathbf{7.51 \pm 1.52}$ **3.21**
Smile	APE:12.47 SNR:0	9.45 ± 2.39 3.03	10.83 ± 2.56 2.57	8.57 ± 2.11 2.34	23.26 ± 9.17 1.29	$\mathbf{7.70 \pm 1.30}$ **3.16**
Surprise	APE:12.29 SNR:0	10.37 ± 2.04 2.61	11.83 ± 2.40 2.27	9.38 ± 1.88 2.08	22.62 ± 8.41 1.32	$\mathbf{8.52 \pm 1.33}$ **2.72**
Squint	APE:12.09 SNR:0	9.48 ± 2.28 2.91	11.01 ± 3.94 2.50	8.62 ± 2.04 2.26	23.17 ± 9.35 1.32	$\mathbf{7.77 \pm 1.50}$ **3.05**
Disgust	APE:12.34 SNR:0	9.82 ± 2.45 2.86	11.11 ± 2.49 2.44	8.96 ± 2.22 2.21	26.07 ± 9.02 1.20	$\mathbf{7.95 \pm 1.53}$ **2.97**
Scream	APE:12.57 SNR:0	10.98 ± 2.70 2.54	12.49 ± 2.70 2.19	9.88 ± 2.40 2.04	25.40 ± 9.61 1.22	$\mathbf{8.77 \pm 1.43}$ **2.69**

methods reconstruct the test image purely as a combination of a training set, the noisy test image is over-smoothed (indicated by a higher SNR) and the person-specific characteristics such as glasses, beard, teeth and wrinkles on the faces, are typically lost. Nguyen and De la Torre [17] did a better job in preserving the subtle visual features on face and re-

sults in lower APE. But it added some noise which lowered the SNR compared to Mika et al., Kwok and Tsang's and our methods. Honeine and Richard's method showed poor performances in terms of both APE and SNR. This is because it assumes a global linear structure between the input space and the feature space, which does not hold in practice. Finally, our method has significantly lower pixel error, higher SNR, and keeps photo-realistic features of the clean image. On the other hand, our method removes the noise while preserving the subtle person-specific (or test-specific) characteristics. Table 1 summarizes the quantitative performance of all methods. Our method outperforms the others in both the APE and SNR criterion.

6. Conclusion

This paper proposes a novel framework to solve the pre-image problem by formulating a local isomorphism between local Gram matrices in the input space and the feature space induced by a kernel. This local isomorphism allows us to establish a bi-directional mapping between the two spaces using second-order statistics. We illustrate the benefit of our approach with two problems: finding the pre-image using non-differentiable kernels and test-data-specific image denoising. More importantly, our framework elegantly overcomes the limitations of previous methods to handle the non-differentiable and application-specific kernels, and there is a closed-form solution. It is important to notice that most of the existing state-of-the-art visual features such as HOG, DAISY or SIFT are non-linear and non-differential operators in the images, that will induce non-differentiable kernels. Both qualitative and qualitative evaluations illustrate that our algorithm outperforms state-of-the-art methods for solving the pre-image problem.

Acknowledgements

The first author was partially supported by National Science Foundation of China: Vision Cognition and Intelligent Computation on Facial Expression under Grant 60973070.

References

[1] M. Aizerman, E. Braverman, and L. Rozonoér. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964. 2761

[2] P. Arias, G. Randall, and G. Sapiro. Connecting the out-of-sample and pre-image problems in kernel methods. *CVPR*, 2007. 2762

[3] G. Bakir, J. Weston, and B. Schölkopf. Learning to find pre-images. *NIPS*, 2003. 2762

[4] Y. Bengio, J. Paiement, and P. Vincent. Out-of-sample extensions for lle, isomap, mds, eigenmaps and spectral clustering. *NIPS*, 2004. 2762, 2763

[5] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. *In ACM Fifth Annual Workshop on Computational Learning Theory*, 1992. 2761

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, pages 886–893, 2005. 2766

[7] I. Dhillon, Y. Guan, and B. Kulis. Kernel k-means: spectral clustering and normalized cuts. *International Conference on Knowledge Discovery and Data Mining*, 2004. 2765

[8] R. Fisher. The statistical utilization of multiple measurements. *Annals of Eugenics*, 8:376–386, 1938. 2761

[9] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. The cmu multi-pose, illumination, and expression (multi-pie) face database. *Tech. rep., Robotics Institute, Carnegie Mellon University,TR-07-08*, 2007. 2765, 2766

[10] P. Honeine and C. Richard. Solving the pre-image problem in kernel machines: a direct method. *Proc. of the 19th IEEE Workshop on Machine Learning for Signal Processing*, 2009. 2762, 2765, 2766, 2767

[11] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936. 2761

[12] I. Jolliffe. *Principal Component Analysis, New York: Springer-Verlag*, 1986. 2761

[13] J. Kwok and I. Tsang. The pre-image problem in kernel methods. *IEEE Trans. on Neural Networks*, 15(6):1517–1525, 2004. 2762, 2764, 2765, 2766, 2767

[14] C. H. Lampert. Kernel methods in computer vision. *Found. Trends. Comput. Graph. Vis.*, 4:193–285, 2009. 2761

[15] N. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *NIPS*, 2004. 2763

[16] S. Mika, B. Schölkopf, A. Smola, K. Müller, M. Scholz, and G. Rätsch. Kernel pca and de-noising in feature spaces. *NIPS*, 1999. 2762, 2764, 2765, 2766, 2767

[17] M. Nguyen and F. De la Torre. Robust kernel principal component analysis. *NIPS*, 2008. 2762, 2764, 2766, 2767

[18] Y. Rathi, S. Dambreville, and A. Tannenbaum. Statistical shape analysis using kernel pca. *In SPIE. Symposium on Electronic Imaging*, 2006. 2762

[19] B. Schölkopf and A. Smola. Nonlinear component analysis as a kernel eigenvalue problem. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press*, 2002. 2761

[20] B. Schölkopf, A. Smola, and K. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998. 2761

[21] W.-S. Zheng, J. Lai, and P. Yuen. Weakly supervised learning on pre-image problem in kernel methods. *ICPR*, 2006. 2762