A Novel Kernel PCA/KLT Approach for Transform Coding of Waveforms

Thomas Sikora
Communication Systems Lab
Technische Universität Berlin
sikora@nue.tu-berlin.de

***Abstract*— A novel Kernel PCA/Kernel KLT transform (S-KPCA) is introduced which incorporates higher order statistics into the design of the transform matrix using a Reproducing Kernel Hilbert Space (RKHS) approach. The goal is to arrive at an orthonormal transform matrix *E* with column eigenvectors that allow reconstruction of an input vector with few coefficients and superior signal fidelity. In contrast to the well known Kernel PCA the number of the generated transform coefficients is not dependent on the size of the training set and the "pre-image problem" is avoided completely. Results indicate that the derived transform is more compact than the standard PCA/KLT in terms of fidelity measures in RKHS.**

I. INTRODUCTION

The Karhunen-Loeve Transform (KLT), also called Principal Component Analysis (PCA), is an established means for transforming a vector x into a compact set of uncorrelated coefficients using a linear transform matrix E [1][2]. The compactness of the transform makes the KLT/PCA approach attractive for many applications. Often the goal is to reconstruct the elements in vector x using a subset of the transform coefficients in c and the associated eigenvector/basis functions. Analyzing the variance of the coefficients allows to select a few basis functions and coefficients of the transform for reconstruction. This is a classical approach used for dimensionality reduction of transformed feature vectors c in signal classification tasks. In information theory and practical communications systems the KLT/PCA is often used as a pre-processing step for "pre-whitening" a signal [1]. In image and video coding standards such as JPEG, MPEG-1/2/4 and MPEG AC3 audio coding, variants of the above PCA/KLT transform coding approach are well established [3]. In particular the well known Discrete Cosine Transform (DCT) and related variants were derived from the optimal PCA/KLT.

The PCA/KLT is designed based on 2nd order statistics of the random vector source X. As such it is the the optimal transform if the random vector X is multi-dimensional Gaussian distributed. In this case the coefficients are not only pair-wise uncorrelated but even statistically independent [1]. In essentially all practical applications, however, the probability distribution of the random vector X is multimodal and far from Gaussian - PCA/KLT performs suboptimal in this case. During the last 20 years attemps have been made to design transforms by incorporating knowledge about higher order statistical dependencies between elements in X. As an example the Independent Component Analysis (ICA) can be designed by minimizing i.e. the neg-entropy of the transform coefficients. The thus designed transform lacks compactness, which is the most important requirement in transform coding applications. A novel and very intriguing approach is the Kernel PCA [5], which promises interesting results for feature selection, feature reduction and denoising applications. Kernel PCA is designed by diagonalizing the Kernel Gram matrix. The resulting transform is also not necessarily compact (or sparse) [6]. A further serious drawback is that the dimension M of the coefficient vector c is dependent on the size M of the training data set. The Kernel PCA usually arrives at a heavily overdetermined set of basis functions. In addition, the well-known "pre-image problem" for reconstructing vector x based on a subset of transform coefficients causes significant difficulties and computational burden for kernels except the Gaussian kernel [6]. Consequently, not much work has been reported thus far on adaption of Kernel PCA for transform coding of signals.

In this paper a novel approach for Kernel PCA/KLT transform is proposed. The novel transform incorporates higher order statistics into the design of the basis functions using a kernel approach - but significantly departs from the standard Kernel PCA approach. The transform arrives at a complete, non-overdetermined set of basis functions in transform matrix E and consequently the number of transform coefficients in c is equal to the number of elements in vector x, as with PCA/KLT. The computational demand for designing the basis functions of the kernel transform is not significantly higher than that of the classical PCA/KLT approach. Most importantly, the transform is compact and outperforms the conventional PCA/KLT in terms of compactness in RKHS.

II. DESCRIBING PROBABILITY DISTRIBUTIONS IN RKHS FEATURE SPACE

Our strategy for the design of the Kernel Transform is to describe the statistical dependency/similarity between zero-mean random elements X_i and X_j in a zero-mean random vector X in terms of the covariance $E\left[\varphi(X_i)^T \cdot \varphi(X_j)\right]$ between the feature vectors $\varphi(X_i)$ and $\varphi(X_j)$, $\varphi(.) \in R^N$. The feature vectors capture non-linear components of X_i and X_j and their covariance measures similarity in RKHS. The covariance matrix $C_{\varphi(X)\varphi(X)}$ captures the covariance between all L elements $\varphi(X_i)$ and $\varphi(X_j)$ in vector X and is of size (LxL).

Recall, that in contrast the standard PCA/KLT explores the second order statistics by employing the covariance terms $E[X_i \cdot X_j]$ for covariance matrix C_{XX} with size (*LxL*).

In general, given two random variables *X* and *Y* in Hilbert space we can describe the characteristics of the density *p(X)* and *p(Y)* in terms of moments. Towards this end we form a nonlinear expansion of *X* and *Y* using the expansion vector $\varphi(X)$ and $\varphi(Y)$. The nonlinear expansion vector captures the nonlinear components of the random variables. We will restrict ourselves to non-linear expansions that transform our random input variable from Hilbert space into the Reproducing Kernel Hilbert Space (RKHS) [5]. The RKHS is a metric space of possibly infinite dimension [5][7]. As such it is possible to calculate normed distances between $\varphi(X)$ and $\varphi(Y)$ for our Kernel transform using inner products – identical as in Hilbert space for the design of the PCA/KLT.

We consider $\varphi(.)$ to be a feature vector of a Mercer Kernel *k(.,.)* and $\varphi(.)$ may be of infinite dimension. Notice that we are not restricted to scalar random variables and *X* may be a vector random variable in R^L. Since *X* and *Y* are random variables also the vectors $\varphi(X)$ and $\varphi(Y)$ are random vector variables and the components are random entries. According to the Mercer theorem an inner product between two feature vectors of the same kernel can be evaluated through the Mercer kernel function $\varphi(X)^T \cdot \varphi(Y) = k(Y,X)$. We can thus evaluate the inner product between features that life in a possibly infinite dimensional RKHS space without the need for calculating the inner product explicitly – by evaluating the kernel function.

The density functions of *p(X)* and *p(Y)* can be fully characterized by the so-called „mean embedding“ vectors $\mu_X = E[\varphi(X)]$, $\mu_Y = E[\varphi(Y)]$ if $\varphi(X)$ and $\varphi(Y)$ are feature vectors of a so-called „characteristic" kernel function *k(.,.)* [7]. Notice that not every Mercer kernel is also a "characteristic" kernel. Popular examples of characteristic kernels are the Laplacian kernel and the Gaussian kernel. In this paper we will explore the Kernel Transform using the translation invariant Gaussian kernel $k(X,Y) = e^{-B(X-Y)^2}$. We note, however, that the approach is not restricted to this kind of kernel function. The kernel can be factorized into the following form:

$$k(X,Y) = k(X-Y) = e^{-B(X-Y)^2} = e^{-B\cdot X^2} \cdot e^{-B\cdot Y^2} \cdot \sum_{n=0}^{N=\infty} \frac{(2B)^n (X^T Y)^n}{n!} = \varphi^T(X) \cdot \varphi(Y)$$

$$\varphi^T(X) = e^{-B\cdot X^2}\left[1 \quad 2B\cdot X \quad 2B^2\cdot X^2 \quad \frac{4}{3}B^3\cdot X^3 \ldots \quad \ldots\right]$$

$$\varphi^T(Y) = e^{-B\cdot Y^2}\left[1 \quad 2B\cdot Y \quad 2B^2\cdot Y^2 \quad \frac{4}{3}B^3\cdot Y^3 \ldots \quad \ldots\right]$$

are the feature expansion vectors of *X* and *Y*, both of infinite dimensions $N = \infty$. The inner product can be evaluated through the kernel function $k(X,Y) = k(X-Y) = \varphi^T(X) \cdot \varphi(Y)$.

The embedding captures weighted „moments“ of the distribution *p(X)*. The embedding is "injective" – for each density distribution function *p(X)* and *p(Y)* a unique point in (the possibly infinite dimensional) RKHS is identified [7]. Notice that the mean embedding vector itself is not a probability density distribution. For a given pdf, each different kernel type captures different forms of „moments“ - those usually do not coincide with the definitions of skew, kurtosis, etc. Also: the mean embedding depends on the parameters of the kernel, i.e. the mean embedding of a random variable using the Gaussian kernel is dependent on the bandwidth *B*. Since the design of our Kernel Transform will be based on the covariance terms $E[\varphi(X_i)^T \cdot \varphi(X_j)]$ between feature vectors of scalar random variables X_i and X_j, the covariance terms using the Gaussian invariant kernel are now defined as:

$$E[\varphi(X_i)^T \cdot \varphi(X_j)] = E[k(X_i - X_j)] =$$

$$= E\left[e^{-2B(X_i - X_j)}\right] = \sum_{n=0}^{N=\infty} \frac{(2B)^n E\left[\left\{e^{-B\cdot X_i^2} \cdot X_i^T X_j \cdot e^{-B\cdot X_j^2}\right\}^n\right]}{n!}$$

It is apparent from the above, that for any type of characteristic Mercer kernel the covariance $E[\varphi(X_i)^T \cdot \varphi(X_j)] = E[k(X_i - X_j)]$ between any two elements in the vector *X* measures the similarity by averaging weighted joint moments of X_i and X_j. Higher order statistical dependencies are thus incorporated into the design of the Kernel Transform.

III. THE KERNEL KLT/PCA TRANSFORM

A. KLT/PCA

Given the zero-mean random input vector variable $X^T = [X_1 \; . \; . \; X_L]$ of size *L*, the standard KLT/PCA diagonalizes the covariance matrix C_{XX}. Here, the covariance terms are defined by the expected values of the outer vector product $X \cdot X^T$. To derive PCA transform matrices *E* the *L* eigenvectors v_i and eigenvalues λ_i of the covariance matrix are obtained by solving the eigenequations $\lambda_i \cdot v_i = C_{XX} \cdot v_i$. All eigenvectors v_i are entries into columns of the matrix *E* and all eigenvalues λ_i into the diagonal matrix Δ, such that $\Delta = E^T \cdot C_{XX} \cdot E$. A particular data input vector *x* is transformed into a PCA coefficient vector using the orthonormal eigenvector matrix *E* by $c = E^T \cdot x$. The eigenvalues are the variances of the coefficients contained in random vector c. Since the covariance matrix Δ of the coefficients is diagonal, the coefficients are pair wise uncorrelated. We recall that the linear transformation

$c = E^T \cdot x$ is energy preserving $\sum_{i=1}^{L} E[c_i^2] = \sum_{i=1}^{L} E[x_i^2]$ and preserves the Shannon entropy of the source vector *X*, *H(X)=H(C)* [1].

B. Kernel PCA

The basic idea related to Kernel PCA as introduced by Schölkopf et al [5] is the expansion of the input data vector *x* into a high dimensional space using feature vectors of Mercer kernels. Once the covariance matrices are constructed or estimated these matrices are diagonalized using the above eigenvector and eigenvalue approach. We have *i=1...M* samples x^i of random vector source *X*, $x^i \in R^L$, available as training data to construct the Kernel PCA transform matrix. In contrast to PCA the non-linear Kernel PCA approach diagonalizes the so-called Kernel Gram matrix of the sample vectors. To this end each measured data vector is transformed into a feature vector $x^i \rightarrow \varphi(x^i)$. The Kernel Gram matrix *G* is constructed by employing inner products between all feature vectors:

$$G_{\varphi\varphi} = \begin{bmatrix} k(x^1,x^1) & . & . & k(x^1,x^M) \\ . & . & . & . \\ . & . & . & . \\ k(x^M,x^1) & . & . & k(x^M,x^M) \end{bmatrix}$$

We assume that the Mercer theorem applies. Since we have *M* data samples of *X*, the size of the matrix is (*MxM*). The eigenvalue equation $\lambda_i \cdot v_i = G_{\varphi\varphi} \cdot v_i$ is solved which involves the calculation of *M* eigenvectors and *M* eigenvalues. Notice, that the Kernel PCA approach thus generates as many transform coefficients as there are data sample vectors in the training set (*M* coefficients). The amount of training samples is usually very high (*M>>L*), which makes this approach not directly suitable for coding applications – since the coefficients (or subsets thereof) need to be coded and transmitted/stored. Additional problem is the "flat" distribution of the variances of the coefficients – the Kernel PCA is usually not compact. In addition the "pre-image" problem, which handles the reconstruction of the data in Hilbert space, is in general an ill-posed problem [6]. The Kernel PCA also does not preserve the Shannon entropy.

C. The Proposed Kernel KLT/PCA Transform

Rather than expanding a random vector variable $X^T = [X_1 \quad X_2 \quad ... \quad X_L]$ using the non-linear expansion $\varphi(X)$ we expand each component X_l, which results in a feature matrix of possibly infinite dimension, $\Psi(X) = [\varphi(X_1) \quad \varphi(X_2) \quad ... \quad \varphi(X_L)]$. The covariance matrix $C_{\varphi\varphi} = E[\Psi^T(X) \cdot \Psi(X)]$ is a *LxL* matrix and captures the desired higher order moments between the vector elements. By virtue of the Mercer theorem this matrix can be calculated using the kernel functions even if the features are of infinite dimensions. A consistent and efficient estimator using *M* vector samples $x^{jT} = [x_1^j \quad x_2^j \quad ... \quad x_L^j]$ from measured data is given by:

$$\hat{C}_{\varphi\varphi} = \frac{1}{M} \sum_{j=1}^{M} \begin{bmatrix} k(x_1^j,x_1^j) & k(x_1^j,x_2^j) & . & . & k(x_1^j,x_L^j) \\ k(x_2^j,x_1^j) & k(x_2^j,x_2^j) & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ k(x_L^j,x_1^j) & . & . & . & k(x_L^j,x_L^j) \end{bmatrix}$$

The transform matrix *E* is derived by calculating the eigenvectors of $\hat{C}_{\varphi\varphi}$. Notice, that the number of eigenvectors generated is – as desired - not dependent on the number *M* of data samples used for training the matrix.

Let *E* and Λ be the matrices containing the eigenvectors and eigenvalues respectively of $\hat{C}_{\varphi\varphi}$. For coding purposes our task is to transmit input vector x to the receiver. The input to the coding approach is the vector $x^T = [x_1 \quad x_2 \quad ... \quad x_L]$. However, our strategy using the derived Kernel Transform now involves the prior transformation of the data input vector into a feature vector matrix $\Psi(x) = [\varphi(x_1) \quad . \quad . \quad \varphi(x_L)]$ (size *LxN*), N being the dimension of each feature vector. This vector matrix is then transformed into a coefficient matrix *Co* (*LxN*) using the eigenvector matrix *E*, $Co = \Psi(x) \cdot E$.

Notice, that the *i'*th column of *Co*, coefficient vector Co_i is the weighted sum of the input feature vectors $Co_i = \Psi(x) \cdot e_i = \sum_{k=1}^{L} e_{i,k} \cdot \varphi(x_k)$. Our developed strategy would thus involve calculating feature matrices of possibly infinite dimensions and to code/transmit coefficient matrices *Co* of possibly infinite dimensions. Since this is not a feasible approach we use the Mercer theorem to kernelize the feature input matrix. Consider a set of feature vectors $\varphi^T(c_l)$ expanding arbitrarily chosen centers c_l. Rather than coding coefficient vector Co_i we may encode the scalar coefficient

$$\gamma_i = \left\{ \sum_{l=-\infty}^{\infty} \alpha_l \cdot \varphi^T(c_l) \right\} Co_i = \sum_{k=1}^{L} e_{i,k} \cdot \sum_{l=-\infty}^{\infty} \alpha_l \cdot k(c_l, x_k)$$

By using a small number of appropriately chosen non-zero coefficients α_l a suitable coding approach is derived that allow transmission of scalar coefficients γ_i to the receiver. For *i=1...L* the matrix equation $\gamma = E^T \cdot k$ is constructed with

$k^T = \left[\sum_{l=-\infty}^{\infty}\alpha_l \cdot k(c_l, x_1), ..., \sum_{l=-\infty}^{\infty}\alpha_l \cdot k(c_l, x_L)\right]$ as input vector. The elements of x are thus "kernelized" prior to transformation.

In a coding scenario we transmit coefficient vector γ and reconstruct k at the receiver. However, the final goal is to convey the amplitudes of vector x from k, which is feasible using appropriate choice of α_l and c_l. One particular choice seems attractive for our purposes: we recognize that the term $\sum_{l=-\infty}^{\infty}\alpha_l \cdot k(c_l, x_k)$ can be designed to approximate each x_k with arbitrary precision, such that $\sum_{l=-\infty}^{\infty}\alpha_l \cdot k(c_l, x_k) = x_k$. What follows is that now we can use vector x directly as input to our liner transform, that is: $\gamma = c = E^T \cdot x$. This is a somewhat unexpected but very fortunate design option using the kernelization approach. We do not have to kernelize the input vector x in order to perform a transform using the Kernel Transform. In particular we do not have to recover x from k at the receiver. The transform matrix E replaces the one that would have been derived using the standard PCA approach on a conventional covariance matrix. We simply compute the eigenvectors based on another, more suitable covariance Toeplitz matrix. Everything else in the transform coding scenario remains the same. We stress that the resulting orthonormal Kernel Transform $c = E^T \cdot x$ is, as with PCA, energy preserving and preserves the Shannon entropy of the source vector.

IV. Performance Evaluation

In order to evaluate the performance of the suggested Kernel Transform approach (S-KPCA) M=1000 training data sample vectors of various length L were captured from horizontal scan lines of test images. Here we report on results for the test images "Lena" and "MRI". An equal amount of sample data vectors were taken from the images and used for testing. For both S-KPCA and standard PCA/KLT the appropriate covariance matrices were constructed and respective transform matrices E calculated. Different matrices E were generated for different length L of the input vectors. For generation of the covariance matrix of the S-KPCA approach the above invariant 1-D Gaussian kernel was used with bandwidth B=0.0006. As expected the choice of the bandwidth has a significant impact on the performance of the S-KPCA. However, the selected bandwidth proofed sufficient for the evaluation at hand. The choice of the kernel seems less important.

How do we evaluate the capability of a transform for reconstructing the fidelity of a signal x - using a sub-set of coefficients? Since we are interested in reconstructing statistical information including higher order moments, the traditional 2nd order statistics means-squared-error approach (or using the coding gain [1]) would not provide any insight into the capability of S-PCA. PCA would always outperform S-KPCA using such measures.

Figure 1 illustrates the capability for reconstructing the 1-D probability density function $p(X_i)$ of an individual element in vector X. $p(X_i)$ was calculated using a 1-D Kernel density estimate with a Gaussian kernel. The S-KPCA design approach attempts to optimize the capability of each transform coefficient - with the goal to providing a better fidelity contribution than PCA for reconstruction of the L-dimensional joint pdf $p(X)$.

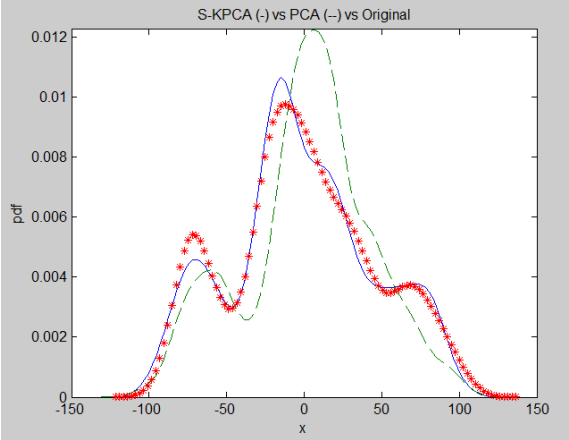

Figure 1: Reconstructed 1-dimensional pdf $p(X_1)$ of the first element in vector X based on a subset of coefficients with highest variance (test image "Lena"), L=128.

Visual inspection of $p(X_1)$ (of the first element of vector X) reveals that using S-KPCA on test image "Lena" (here as an example with L=128) the reconstruction with the first 8 out of 128 coefficients of highest variances already provides support for the three main modes of the desired original distribution. PCA requires significantly more coefficients to achieve this goal. The result in Figure 1 already provides insight into the capability of the S-KPCA approach. However, only a separated element X_i was captured and it is possible that other elements do not benefit from the approach or even suffer. Here we propose to employ a fidelity criterion that measures the distance between the reconstructed L-dimensional density function (of reconstructed random vector X) and the original distribution of X. Traditional criteria include the Kulback-Leibler measure, Kolmogorov distance and the like [8] but require explicit estimation of the L-dimensional density. It is known that approaches like the above Kernel Density Estimation are not capable of dealing with higher-dimensional distributions sufficiently.

With the introduction of the kernel embeddings, however, it is possible to measure distances in RKHS without explicit density estimation. The so-called Minimum Mean Distance (*mmd*) between two mean embeddings is defined as [7]

$$mmd = \left\| \mu_x - \mu_Y \right\|^2 = \mu_X^T \mu_x + \mu_Y^T \mu_Y - 2\mu_X^T \mu_Y$$
$$\approx 1/M^2 \left\{ \sum_{i,j=1}^{M} k(x_i, x_j) + \sum_{i,j=1}^{M} k(y_i, y_j) - 2\sum_{i,j=1}^{M} k(x_i, y_j) \right\}$$

and measures the difference between L-dimensional distributions $p(X)$ and $p(Y)$ in RKHS. mmd=0 if $p(X)$ and $p(Y)$ are identical. A sufficient estimator of mmd uses M samples of vector X and kernelizes the mean embeddings accordingly. Figure 2 depicts the mmd measures for S-KPCA and PCA for test images "Lena" and "MRI".

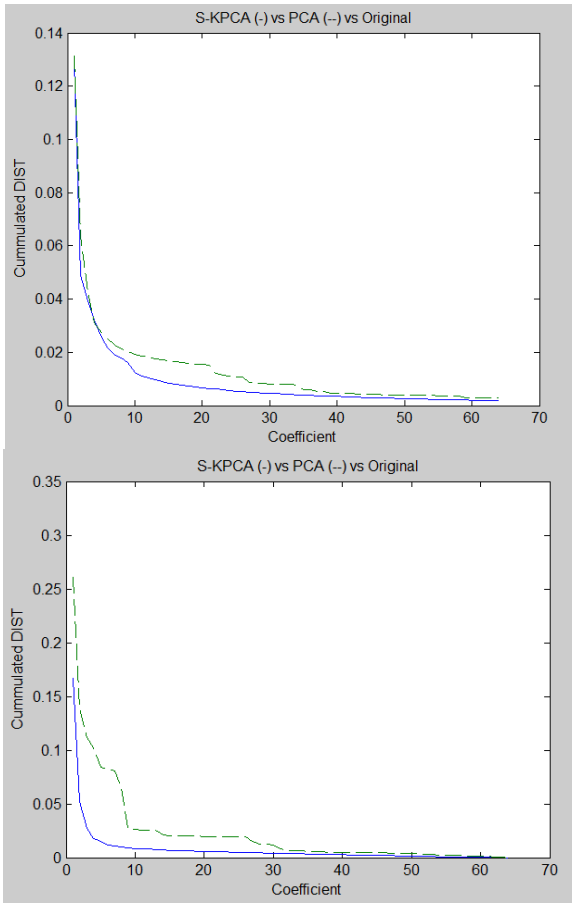

Figure 2: Top: Cummulated mmd distance in RKHS between L-dimensional pdf $p(X)$ of the reconstructed of vector X and the pdf of the original. "Lena" (top) and "MRI" (bottom). ___ S-KPCA, _ _ PCA, * Original.

It is apparent that both S-KPCA and PCA are also compact in terms of mmd using accumulated coefficients. S-KPCA for images "MIR" and "Lena" significantly outperform PCA based on the mmd measure. Similar and consistent results were obtained for other test images and for different length L of the transform. Notice, that mmd measures distances in L-dimensional RKHS, with L=128 and L=64 respectively. Further results not presented in this paper indicate that S-KPCA also preserves the compactness of PCA in regard to the variances of the coefficients. This is a good indication that S-KPCA coefficients may be coded with excellent rate-mmd performance

V. Summary, Conclusion and Outlook

The novel Kernel PCA/Kernel KLT transform (S-KPCA) introduced incorporates higher order statistics into the design of the transform matrix using a Reconstructing Kernel Hilbert Space (RKHS) approach. In contrast to the well-known Kernel PCA the number of the generated transform coefficients of the suggested approach is not dependent on the size of the training set and the "pre-image problem" is avoided completely. First results indicate that the derived transform is more compact than the standard PCA/KLT in terms of the mmd fidelity measure in RKHS. The 1-dimensional S-KPCA transform introduced above can be readily extended towards higher-dimensional transforms in image and video processing and coding. The large variety of kernels available in literature allows the employment of S-KPCA for compression purposes far beyond continuous amplitude sources – examples include compression of text, binary sources, etc. Even though the introduced Kernel Transform is mainly discussed in the context of transform coding applications, it is understood that potential applications are in all fields covered by the traditional PCA, including feature dimensionality reduction, noise reduction and the like.

References

[1] N. Jayant and P. Noll, "Digital Coding of Waveforms," John Wiley & Sons, 1998.

[2] A. Hyvärinen J. Karhunen, E. Oja, "Independent Component Analysis", Prentice Hall, 1984.

[3] T. Sikora, "Trends and Perspectives in Image and Video Coding", Proceedings of the IEEE (Volume: 93 , Issue: 1), 2005.

[4] I. Jolliffe, "Principal Component Analysis", John Wiley & Sons, 2005.

[5] B. Schölkopf, A. Smola, K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem", Neural Computation 10 (5), 1299-1319, 1998.

[6] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, G. Rätsch, "Kernel PCA and De-Noising in Feature Spaces", Proc. NIPS, 1998.

[7] L. Song, L. Fukumizu, K. and A. Gretton, A., "Kernel Embeddings of Conditional Distributions: A Unified Kernel Framework for Nonparametric Inference in Graphical Models", IEEE Signal Processing Magazine, Vol. 30, 2013.

[8] S. K. Zhou, R. Chellappa, "From Sample Similarity to Ensemble Similarity: Probabalistic Distance Measures in Reproducing Kernel Hilbert Space", Proc. IEEE Pattern Analysis and Machine Intelligence", Vol. 28, June 2006.