Incremental Slow Feature Analysis with Indefinite Kernel for Online Temporal Video Segmentation

Stephan Liwicki[1], Stefanos Zafeiriou[1], and Maja Pantic[1,2]

[1] Department of Computing, Imperial College London, United Kingdom
{sl609, s.zafeiriou, m.pantic}@imperial.ac.uk
[2] Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, The Netherlands

Abstract. Slow Feature Analysis (SFA) is a subspace learning method inspired by the human visual system, however, it is seldom seen in computer vision. Motivated by its application for unsupervised activity analysis, we develop SFA's first implementation of online temporal video segmentation to detect episodes of motion changes. We utilize a domain-specific indefinite kernel which takes the data representation into account to introduce robustness. As our kernel is indefinite (*i.e.* defines instead of a Hilbert, a Krein space), we formulate SFA in Krein space. We propose an incremental kernel SFA framework which utilizes the special properties of our kernel. Finally, we employ our framework to online temporal video segmentation and perform qualitative and quantitative evaluation.

1 Introduction

Slow Feature Analysis (SFA) is an unsupervised technique for dimensionality reduction, which extracts slowly varying features from rapidly changing raw input signals [1]. The intuition behind SFA is linked to the assumption that the information (*e.g.* activities or actions) contained in a signal (*e.g.* a video) do not change suddenly, but slowly over time. While the input signal has generally high variation (*e.g.* due to noise) the separation between informative changes is usually hidden in the seldom varying features of the sequence. SFA extracts such features, as it selects the attributes of the video which change least over time.

Although SFA only recently found its way into the computer vision community [2–5], it is commonly linked to the visual cortex [6, 1]. In [2], its properties are exploited to segment videos temporally. The individual segments are thought to be the activities in the video. After performing SFA on the complete video, they determine whether a split of the sequence is required. The decision is based on the median of change in the slow features. For the separation, the frame with the largest change is utilized as split position. Another set of SFA is performed on the resulting videos, and the process is repeated until no further split is necessary. For this, however, the complete video must be known *a priori*.

When the images are from a live camera stream, however, the complete video is unknown. If SFA is used for each time-step, an incremental learning algorithm

is required. Closely related to incremental SFA is incremental Principal Component Analysis (PCA) [7, 8], as SFA can be solved *via* PCA and Minor Components Analysis (MCA). One incremental version of SFA (IncSFA), which utilizes Candid Covariance-Free Incremental PCA (CCIPCA) [9], and MCA is proposed in [3]. Due to the nature of CCIPCA, IncSFA only learns an estimation of the real features, and it requires many training examples (epochs) to converge to the real solution. Thus it is fast, but not exact.

Another important aspect of SFA is the choice of data representation. Originally, SFA was designed for linear data sets which also allow for non-linear expansions, such as a quadratic expansion [1]. More recently, [5] introduce Kernel SFA (KSFA) with kernels in Hilbert space. Often, the computation of a kernel is more efficient than non-linear expansions. A typical choice, also taken in [5], is the selection of standard kernels, such as Gaussian RBFs (GRBFs). However, such kernels have drawbacks: (1) standard kernels seldom utilize the domain dependent property of the data, and (2) common problems exist when an incremental learning method is required (*e.g.* the construction of a reduced set representation). Typically, the online classification of an online kernel learning method is written as a weighted sum of kernel combination of samples from a set of stored instances, usually referred to as support or reduced set. At each step a new instance is fed to the algorithm and depending on the update criterion the algorithm adds the instance to the support set. A major challenge in online learning is that the support set may grow arbitrarily large over time [10].

An incremental kernel PCA (KPCA) algorithm which kernelizes the exact algorithm for incremental PCA in [7, 8] (IPCA) is proposed in [10]. In this method, in order to maintain constant update speed the authors construct a reduced set expansion, by means of pre-images. The main drawbacks of this method are that (1) the reduced set representation provides only an approximation to the exact solution and (2) the proposed optimization problem for finding the expansion inevitably increases the complexity of the algorithm.

Some related works in the broader area of unsupervised video segmentation are [11–13]. In [11] a method for clustering facial events is proposed. Their work is only suitable for offline processing and requires the number of clusters *a priori*. This is also the case for the clustering algorithm in [12]. A method for joint segmentation and classification of human actions in video is proposed in [13]. Their method is supervised, *i.e.* a model for human actions is learned from a set of labeled training samples. Then, given a testing video with a continuous stream of human activities, the algorithm in [13] finds the globally optimal temporal segmentation (*i.e* the change points between actions) and class labels.

Our methodology takes a different direction. In particular, we detect the temporal changes in video streams online. We do not require the number of clusters, nor train to a predefine set of examples. Thus, the methods in [11–13] constitute excellent post-processing tools for clustering or classifying the events.

In this paper, we propose an online algorithm for incremental SFA which builds on a special kernelized version of IPCA. The original KSFA [5], described above, only supports arbitrarily chosen positive definite kernels. We take a differ-

ent route. Rather than using off-the-shelf kernels which do not incorporate any problem-specific prior knowledge, we utilize our kernel presented in [14] which is based on a modification of a gradient-based correlation [15]. In [14], we analyze this kernel and show superior performance for object tracking and recognition. Finally, rather than using CCIPCA for online SFA, we utilize IPCA to produce an exact incremental update of SFA at each time-step.

An important aspect of our framework is that our kernel is not positive definite and, thus, the appropriate space in which our kernel can represent a dot-product is a Krein space. Therefore, we start by formulating KSFA in a Krein space. We then show that our kernel has a very special form which enables us to formulate a direct version of our KSFA which does not require the computation of a reduced set expansion. We capitalize on this property and propose an efficient and exact incremental KSFA with our indefinite kernel. Finally, we develop SFA's first real-time temporal video segmentation algorithm. In summary, our contributions are: (1) We propose KSFA in Krein space as our kernel is indefinite. (2) We formulate incremental SFA using the exact IPCA which produces a close approximation of SFA after each time-step. (3) We propose an accurate incremental KSFA in Krein space which exploits the properties of our kernel and does not require a reduced set expansion. (4) We use our learning framework to implement the first online temporal video segmentation with SFA and validate its performance on several video sequences.

The remainder of this paper is as follows. In Section 2, we describe standard batch SFA for training data known *a priori*. In Section 3, we present our utilized kernel and introduce incremental KSFA in Krein space. The framework for our temporal video segmentation is proposed in Section 4, and our experiments are shown in Section 5. Section 6 concludes our paper.

2 Slow Feature Analysis

Given n sequential observation vectors $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_n]$, SFA finds an output signal $\mathbf{O} = [\mathbf{o}_1 \cdots \mathbf{o}_n]$ for which the features change *slowest* over time [1].

The output of each individual sample is formed as the concatenation of k mappings $y_j, j = 1, \ldots, k$, such that $\mathbf{o}_i = [y_1(\mathbf{x}_i) \cdots y_k(\mathbf{x}_i)]^T$, where $(.)^T$ computes the transpose. SFA minimizes the *slowness* for these values, defined as

$$\Delta(y_j) = \frac{1}{n}\dot{\mathbf{y}}_j^T \dot{\mathbf{y}}_j \tag{1}$$

where $\dot{\mathbf{y}}_j = [\dot{y}_j(\mathbf{x}_1) \cdots \dot{y}_j(\mathbf{x}_n)]^T$ is the sequentially concatenated vector containing the derivatives of y_j for the sequence. Generally, $\dot{y}_j$ is represented as the difference between consecutive time steps, $\dot{y}_j(\mathbf{x_t}) = y_j(\mathbf{x_t}) - y_j(\mathbf{x_{t-1}})$ [5, 4, 2].

Additional constrains are introduced to avoid the trivial solution and prevent information redundancy. The output signals of each $\mathbf{y}_j = [y_j(\mathbf{x}_1) \cdots y_j(\mathbf{x}_n)]^T$ are required to have zero mean, and unit variance. Moreover, all $\mathbf{y}_j$ are constrained to be uncorrelated. Finally, a useful additional constraint is the ordering

of the components. We summarize the constrains in the following

$$\begin{array}{llll} \forall i & \mathbf{y}_i^T \mathbf{1}_{n\times 1} = 0 & \forall i & \mathbf{y}_i^T \mathbf{y}_i = 1 \\ \forall i \neq j & \mathbf{y}_i^T \mathbf{y}_j = 0 & \forall i < j & \Delta(y_i) < \Delta(y_j) \end{array} \tag{2}$$

where $\mathbf{1}_{a\times b}$ is an $a \times b$ matrix with all elements set to 1.

Often, the input features $\mathbf{x}_i$ are assumed to be linear, $\mathbf{z}_i = \mathbf{x}_i$, or a result of a nonlinear expansion, $\mathbf{z}_i = h(\mathbf{x}_i)$ (*e.g.* a quadratic expansion). Then, SFA can be solved by means of the generalized eigenvalue problem [1]:

$$\min_{\mathbf{B}} tr\left(\left(\mathbf{B}^T \mathbf{Z}\mathbf{Z}^T \mathbf{B}\right)^{-1} \mathbf{B}^T \dot{\mathbf{Z}}\dot{\mathbf{Z}}^T \mathbf{B}\right) \tag{3}$$

where $\mathbf{Z} = [\mathbf{z}_1 \cdots \mathbf{z}_n]$ and $\dot{\mathbf{Z}}$ contain the input features and their time derivative respectively, and $tr(.)$ computes the trace of a matrix. As in [1], after finding the whitening matrix $\mathbf{W}$, such that $\mathbf{W}^T\mathbf{Z}\mathbf{Z}^T\mathbf{W} = \mathbf{I}$, eq. (3) can be simplified to

$$\min_{\mathbf{A}} tr\left(\mathbf{A}^T \mathbf{W}^T \dot{\mathbf{Z}}\dot{\mathbf{Z}}^T \mathbf{W}\mathbf{A}\right) \text{ , s.t. } \mathbf{A}^T\mathbf{W}^T\mathbf{Z}\mathbf{Z}^T\mathbf{W}\mathbf{A} = \mathbf{A}^T\mathbf{A} = \mathbf{I}. \tag{4}$$

Note, for simplicity, we compute the matrix of derivations from $\mathbf{Z}$ [5, 4, 2],

$$\dot{\mathbf{Z}} = [\mathbf{z}_2 \cdots \mathbf{z}_n] - [\mathbf{z}_1 \cdots \mathbf{z}_{n-1}] = \mathbf{Z}\mathbf{P}_n \tag{5}$$

where $\mathbf{P}_n$ is an $n\times(n-1)$ matrix with elements $\mathbf{P}_n(i,i) = -1$ and $\mathbf{P}_n(i+1,i) = 1$.

We now briefly outline a batch algorithm which solves SFA when the data is known *a priori*. Given a p_1-dimensional sequential input signal of n samples $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_n] \in \mathbb{R}^{p_1\times n}$, its p_2-dimensional expansion is given by $\mathbf{Z} = [\mathbf{z}_1 \cdots \mathbf{z}_n] \in \mathbb{R}^{p_2\times n}$, where $\mathbf{z}_i = h(\mathbf{x}_i)$ is build using the non-linear mapping $h : \mathbb{R}^{p_1} \to \mathbb{R}^{p_2}$.

Generally, the data in $\mathbf{Z}$ may not have zero mean, however, a centered data matrix can be easily computed. We find the centralized data matrix $\bar{\mathbf{Z}} = \mathbf{Z} - \mathbf{1}_{n\times 1}\boldsymbol{\mu}_{\mathbf{Z}}$, where $\boldsymbol{\mu}_{\mathbf{Z}}$ is the mean of the samples. In the first step, the matrix $\mathbf{W}$ which whitens the signal is calculated, such that $\mathbf{W}^H\bar{\mathbf{Z}}\bar{\mathbf{Z}}^H\mathbf{W} = \mathbf{I}$. For this, we require a singular value decomposition (SVD) of $\bar{\mathbf{Z}}$. For high-dimensional data we find the eigenvalue decomposition (ED) of $\bar{\mathbf{Z}}^T\bar{\mathbf{Z}} = \boldsymbol{\Omega}\boldsymbol{\Lambda}\boldsymbol{\Omega}^T$ and get [16]

$$\bar{\mathbf{Z}} = \left[\bar{\mathbf{Z}}\boldsymbol{\Omega}\boldsymbol{\Lambda}^{-\frac{1}{2}}\right]\left[\boldsymbol{\Lambda}^{\frac{1}{2}}\right][\boldsymbol{\Omega}]^T = \mathbf{U}\mathbf{D}\mathbf{V}^T. \tag{6}$$

The projection which whitens the the scatter matrix is provided by $\mathbf{W} = \mathbf{U}\mathbf{D}^{-1}$.

An optional dimensionality reduction of the input data may be introduced for the whitening projection and the eigenspace, such that $\mathbf{W}_{k_1} = \mathbf{U}_{k_1}\mathbf{D}_{k_1}^{-1}$ where $\mathbf{U}_{k_1}$ and $\mathbf{D}_{k_1}$ correspond to the k_1 largest eigenvalues in $\mathbf{D}$. In the following, we generally omit the subscript which indicates this reduced space.

SFA's second step solves eq. (4). First, the mean $\boldsymbol{\mu}_{\dot{\mathbf{Z}}}$ and the centered data matrix $\dot{\bar{\mathbf{Z}}} = \dot{\mathbf{Z}} - \mathbf{1}_{n\times 1}\boldsymbol{\mu}_{\dot{\mathbf{Z}}}$ are computed. To find the final output functions of the SFA, the ED of $\mathbf{W}^T\dot{\bar{\mathbf{Z}}}\dot{\bar{\mathbf{Z}}}^T\mathbf{W} = \mathbf{A}\mathbf{H}\mathbf{A}^T$ is used [1]. The projection which solves eq. (4) is given by $\mathbf{B} = \mathbf{W}\mathbf{A}$. Thus the output signal of a sample $\mathbf{x}_i$ is given by

$$\mathbf{o}_i = \mathbf{A}^T\left(\mathbf{W}^T(\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{Z}}) - \mathbf{W}^T\boldsymbol{\mu}_{\dot{\mathbf{Z}}}\right) = \mathbf{B}^T\left(\mathbf{z}_i - \boldsymbol{\mu}_{\mathbf{Z}} - \boldsymbol{\mu}_{\dot{\mathbf{Z}}}\right). \tag{7}$$

The ordering, in terms of slowness, of the functions y_j, which construct $\mathbf{o}_i$, is provided by the order of the components in $\mathbf{A}$ which is governed by the eigenvalues in $\mathbf{H}$. The slowest function is related to the smallest eigenvalue and the next larger eigenvalue gives the second slowest function, etc.

3 Incremental Slow Feature Analysis in Krein Space

In this section we present our incremental KSFA in Krein space which is designed to exploit the special properties of our kernel. First, we introduce our kernel. We then present a brief introduction to Krein spaces. The development of KSFA in Krein spaces and the exploitation of the special form of our kernel to formulate a direct version of KSFA for our kernel is then shown. Finally, we propose our incremental KSFA with our indefinite kernel.

3.1 The Robust Gradient-based Kernel

Assume that we are given two images $\mathbf{I}_i \in \mathbb{R}^{n\times m}, i = 1, 2$, with normalized pixel values in range $[0, 1]$. The gradient-based representation of $\mathbf{I}_i$ is defined as

$$\mathbf{G}_i = \mathbf{F}_x \star \mathbf{I}_i + j\mathbf{F}_y \star \mathbf{I}_i \tag{8}$$

where $\mathbf{F}_x$ and $\mathbf{F}_y$ are linear filters which approximate the ideal differentiator in the image's horizontal and vertical axis. Let $\mathbf{x}_i \in \mathbb{C}^d$ be the d-dimensional vector obtained from $\mathbf{G}_i$ in lexicographical order. The gradient correlation is given by

$$s_1(\mathbf{x}_i, \mathbf{x}_j) = \mathbb{R}\left\{\mathbf{x}_i^H \mathbf{x}_j\right\} = \sum_{c=1}^{d} \mathbf{R}_i(c)\mathbf{R}_j(c)\cos\left(\Delta\boldsymbol{\theta}(c)\right) \tag{9}$$

where $\mathbb{R}\{.\}$ extracts the real value of a complex number, $\mathbf{R}_i$ is a vector containing the magnitudes of $\mathbf{x}_i$, $\Delta\boldsymbol{\theta}(c) = \boldsymbol{\theta}_i(c) - \boldsymbol{\theta}_j(c)$ is the difference in the orientations, $\mathbf{R}_i(c)e^{j\boldsymbol{\theta}_i(c)}$ is the polar form of $\mathbf{x}_i(c)$, and $(.)^H$ is the complex conjugate transposition. The correlation in eq. (9) was proposed for robust scale-invariant image matching under the presence of occlusions and large non-overlapping regions [15]. Its robustness stems from the choice of features, but also the utilized correlation.

In [14], we propose a modification of this measure, so that $k(\mathbf{x}_i, \mathbf{x}_j)$ can be expressed as the dot-product of two explicit mappings, $a : \mathbb{C}^d \to \mathbb{C}^{2d}$ and $b : \mathbb{C}^d \to \mathbb{C}^{2d}$, such that $k(\mathbf{x}_i, \mathbf{x}_j) = a(\mathbf{x}_i)^H b(\mathbf{x}_j) = b(\mathbf{x}_i)^H a(\mathbf{x}_j)$, and

$$a(\mathbf{x}_i) = \begin{bmatrix} \frac{\mathbf{R}_i e^{j\boldsymbol{\theta}_i}}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_i^2(c)d}} \\ e^{j\boldsymbol{\theta}_i} \end{bmatrix} \; b(\mathbf{x}_i) = \begin{bmatrix} e^{j\boldsymbol{\theta}_j} \\ \frac{\mathbf{R}_j e^{j\boldsymbol{\theta}_j}}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_j^2(c)d}} \end{bmatrix}, \text{ where } e^{j\boldsymbol{\theta}} = \begin{bmatrix} e^{j\boldsymbol{\theta}(1)} \\ \vdots \\ e^{j\boldsymbol{\theta}(d)} \end{bmatrix}. \tag{10}$$

The kernel's robust properties derive from (1) the use of gradient orientations, (2) the addition of magnitudes, (3) the use of the cosine on the difference of gradient orientations (more details in [14]). Thus, this kernel is suitable for robust image processing. Finally, we emphasize that our kernel is non-positive definite. Consequently, we cannot define an implicit Hilbert feature space. In this case, the space where the kernel represents a dot-product is a Krein space [17].

3.2 Krein Spaces

Krein spaces are important as they produce feature-space representations of dissimilarities and provide us with insights on the geometry of classifiers defined with non-positive kernels [17]. An abstract space $\mathcal{K}$ is a Krein space if there exists an (indefinite) inner product $\langle .,. \rangle_{\mathcal{K}} : \mathcal{K} \times \mathcal{K} \rightarrow \mathbb{C}$ such that [18]:

$$\forall x, y \in \mathcal{K} \;\; \langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{K}} = \langle \mathbf{y}, \mathbf{x} \rangle_{\mathcal{K}}^{C} \tag{11}$$

$$\forall x, y, z \in \mathcal{K}, c_1, c_2 \in \mathbb{R} \;\; \langle c_1 \mathbf{x} + c_2 \mathbf{z}, \mathbf{y} \rangle_{\mathcal{K}} = c_1 \langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{K}} + c_2 \langle \mathbf{z}, \mathbf{y} \rangle_{\mathcal{K}} \tag{12}$$

where $(.)^C$ computes the complex conjugate. $\mathcal{K}$ is composed of two vector spaces, $\mathcal{K} = \mathcal{K}_+ \oplus \mathcal{K}_-$, where $\mathcal{K}_+$ and $\mathcal{K}_-$ describe two Hilbert spaces, for which we denote their corresponding positive definite inner products as $\langle .,. \rangle_{\mathcal{K}_+}$ and $\langle .,. \rangle_{\mathcal{K}_-}$ respectively. Note here, all Hilbert spaces are also Krein spaces, as $\mathcal{K}_-$ may be empty. The decomposition of $\mathcal{K}$ into its two subspaces defines two orthogonal projections: $\mathbf{P}_+$ onto $\mathcal{K}_+$ and $\mathbf{P}_-$ onto $\mathcal{K}_-$, known as fundamental projections of $\mathcal{K}$. Using these projections, $\mathbf{x} \in \mathcal{K}$ can be represented as $\mathbf{x} = \mathbf{P}_+\mathbf{x} + \mathbf{P}_-\mathbf{x}$.

Let $\mathbf{x}_+ \in \mathcal{K}_+$ and $\mathbf{x}_- \in \mathcal{K}_-$ be the projections onto the subspaces $\mathbf{P}_+\mathbf{x}$ and $\mathbf{P}_-\mathbf{x}$ respectively. Then, $\langle \mathbf{x}_+, \mathbf{y}_- \rangle_{\mathcal{K}} = 0$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{K}$. Moreover, $\langle \mathbf{x}_+, \mathbf{y}_+ \rangle_{\mathcal{K}} > 0$ and $\langle \mathbf{x}_-, \mathbf{y}_- \rangle_{\mathcal{K}} < 0$ for any non-zero vectors $\mathbf{x}, \mathbf{y} \in \mathcal{K}$. Hence, $\mathcal{K}_+$ is a positive subspace, while $\mathcal{K}_-$ is a negative subspace. The inner product of $\mathcal{K}$ is defined as

$$\forall \mathbf{x}, \mathbf{y} \in \mathcal{K} \;\; \langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{K}} = \langle \mathbf{x}_+, \mathbf{y}_+ \rangle_{\mathcal{K}_+} - \langle \mathbf{x}_-, \mathbf{y}_- \rangle_{\mathcal{K}_-}. \tag{13}$$

$\mathcal{K}$ has an associated Hilbert space $|\mathcal{K}|$ which can be found *via* the linear operator $\mathbf{J} = \mathbf{P}_+ - \mathbf{P}_-$, called the fundamental symmetry. This symmetry satisfies $\mathbf{J} = \mathbf{J}^{-1} = \mathbf{J}^T$ and describes a Krein space's basic properties. By using eq. (13) and $\mathbf{J}$ the connection between $|\mathcal{K}|$ and $\mathcal{K}$ can be written as a "conjugate":

$$\mathbf{x}^*\mathbf{y} \triangleq \langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{K}} = \mathbf{x}^H \mathbf{J} \mathbf{y} = \langle \mathbf{J}\mathbf{x}, \mathbf{y} \rangle_{|\mathcal{K}|} \tag{14}$$

That is, $\mathcal{K}$ can be turned into its associated Hilbert space $|\mathcal{K}|$ by using the positive definite inner product of the Hilbert space,$\langle .,. \rangle_{|\mathcal{K}|}$, as $\langle \mathbf{x}, \mathbf{y} \rangle_{|\mathcal{K}|} = \langle \mathbf{x}, \mathbf{J}\mathbf{y} \rangle_{\mathcal{K}}$.

We are particularly interested in finite dimensional Krein spaces where $\mathcal{K}_+$ is isomorphic to $\mathbb{C}^p$ and $\mathcal{K}_-$ is isomorphic to $\mathbb{C}^q$. Such a Krein space describes a pseudo-Euclidean space [17]. In particular, the symmetry $\mathbf{J} \in \mathbb{R}^{(p+q)\times(p+q)}$ is given by $\forall i \leq p, \mathbf{J}(i,i) = 1$, $\forall p < i \leq (p+q), \mathbf{J}(i,i) = -1$ and $\forall i \neq j, \mathbf{J}(i,j) = 0$.

Our kernel (Section 3.1) is indefinite, and defines an implicit mapping $\phi : \mathbb{C}^d \rightarrow \mathcal{K}$ into a finite Krein space. Analogously to Hilbert space, our kernel is equivalent to the dot-product in feature space, *i.e.* $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{K}}$.

3.3 Indefinite Kernel Slow Feature Analysis in Krein Space

Let us assume a signal $\mathbf{Z} = [\phi(\mathbf{x}_1) \cdots \phi(\mathbf{x}_n)]$ is given, where $\phi(.)$ is an implicit mapping into Krein space $\mathcal{K}$. We find its derivative as $\dot{\mathbf{Z}} = \mathbf{Z}\mathbf{P}_n$. Let us define the total scatter matrices $\mathbf{S}_{\mathcal{K}} \triangleq \bar{\mathbf{Z}}\bar{\mathbf{Z}}^* = \bar{\mathbf{Z}}\bar{\mathbf{Z}}^H\mathbf{J} = \mathbf{S}_{|\mathcal{K}|}$ and $\dot{\mathbf{S}}_{\mathcal{K}} \triangleq \dot{\bar{\mathbf{Z}}}\dot{\bar{\mathbf{Z}}}^* = \dot{\mathbf{S}}_{|\mathcal{K}|}$, where $\mathbf{S}_{|\mathcal{K}|}$ and $\dot{\mathbf{S}}_{|\mathcal{K}|}$ are the scatter matrices of the associated Hilbert space $|\mathcal{K}|$.

Analogously to KSFA in Hilbert space [5] and KPCA with indefinite kernels [14], we formulate the optimization in eq. (3) for Krein spaces as follows[3]

$$\min_{\mathbf{B}} tr\left(\left(\mathbf{B}^*\mathbf{S}_{\mathcal{K}}\mathbf{B}\right)^{-1}\mathbf{B}^*\dot{\mathbf{S}}_{\mathcal{K}}\mathbf{B}\right) \tag{15}$$

By formulating the projection as a linear combination $\mathbf{B} = \bar{\mathbf{Z}}\tilde{\mathbf{B}}$, eq. (15) becomes

$$\begin{aligned}&\min_{\tilde{\mathbf{B}}}\ tr\left(\left(\tilde{\mathbf{B}}^H\bar{\mathbf{Z}}^H\mathbf{J}\bar{\mathbf{Z}}\bar{\mathbf{Z}}^H\mathbf{J}\bar{\mathbf{Z}}\tilde{\mathbf{B}}\right)^{-1}\tilde{\mathbf{S}}^H\bar{\mathbf{Z}}^H\mathbf{J}\dot{\bar{\mathbf{Z}}}\dot{\bar{\mathbf{Z}}}^H\mathbf{J}\bar{\mathbf{Z}}\tilde{\mathbf{B}}\right)\\ =&\min_{\tilde{\mathbf{B}}}\ tr\left(\left(\tilde{\mathbf{B}}^H\bar{\mathbf{K}}\bar{\mathbf{K}}\tilde{\mathbf{B}}\right)^{-1}\tilde{\mathbf{B}}^H\bar{\mathbf{K}}\mathbf{P}_n\mathbf{M}_{n-1}^T\mathbf{P}_n^T\bar{\mathbf{K}}\tilde{\mathbf{B}}\right)\end{aligned} \tag{16}$$

where $\bar{\mathbf{K}}$ is the centralized kernel matrix of the signal, and $\mathbf{M}_n = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_{n\times n}$.

We need to find a solution $\tilde{\mathbf{B}}$ such that $\tilde{\mathbf{B}}^H\bar{\mathbf{K}}^2\tilde{\mathbf{B}} = I$. We compute the ED $\bar{\mathbf{K}}^2 = \mathbf{\Omega}\mathbf{\Lambda}^2\mathbf{\Omega}^H$, and define $\tilde{\mathbf{B}} = \tilde{\mathbf{W}}\mathbf{A} = \mathbf{\Omega}|\mathbf{\Lambda}|^{-1}\mathbf{A}$, then

$$\tilde{\mathbf{B}}^H\bar{\mathbf{K}}^2\tilde{\mathbf{B}} = \mathbf{A}^H|\mathbf{\Lambda}|^{-1}\mathbf{\Omega}^H\mathbf{\Omega}\mathbf{\Lambda}^2\mathbf{\Omega}^H\mathbf{\Omega}|\mathbf{\Lambda}|^{-1}\mathbf{A} = \mathbf{A}^H\mathbf{A} = \mathbf{I} \tag{17}$$

where $\mathbf{\Lambda}$ has p positive and q negative eigenvalues. Its reduced set is obtained by keeping the k_1 eigenvalues with the largest magnitude. Now, eq. (16) becomes

$$\min_{\mathbf{A}} tr\left(\mathbf{A}^H\tilde{\mathbf{W}}^H\dot{\bar{\mathbf{K}}}\dot{\bar{\mathbf{K}}}^H\tilde{\mathbf{W}}\mathbf{A}\right)\ \text{, s.t. } \mathbf{A}^H\mathbf{W}^*\bar{\mathbf{Z}}\bar{\mathbf{Z}}^*\mathbf{W}\mathbf{A} = \mathbf{A}^H\mathbf{A} = \mathbf{I} \tag{18}$$

where $\dot{\bar{\mathbf{K}}} = \bar{\mathbf{K}}\mathbf{P}_n\mathbf{M}_{n-1}$ – *i.e.* the centered derivative of the kernel matrix. The final projection becomes $\mathbf{B} = \bar{\mathbf{Z}}\tilde{\mathbf{B}} = \bar{\mathbf{Z}}\tilde{\mathbf{W}}\mathbf{A} = \mathbf{W}\mathbf{A}$.

We have shown how SFA is formulated with a general kernel in Krein space. Let us now conclude with a special formulation which allows direct computation using the mappings $a(.)$ and $b(.)$ of our kernel in eq. (10). Similar to [14], we substitute the kernel by utilizing $a(.)$ and $b(.)$ (which are *not* equivalent to the implicit mapping $\phi(.)$). Our projection for our special case is therefore given by $\mathbf{B}_a = \bar{\mathbf{Z}}_a\tilde{\mathbf{B}}$ and $\mathbf{B}_b = \bar{\mathbf{Z}}_b\tilde{\mathbf{B}}$, where $\bar{\mathbf{Z}}_a$ and $\bar{\mathbf{Z}}_b$ are the centered data matrices of $\mathbf{Z}_a = [a(\mathbf{x}_1)\cdots a(\mathbf{x}_n)]$ and $\mathbf{Z}_b = [b(\mathbf{x}_1)\cdots b(\mathbf{x}_n)]$ respectively. In the following, we denote the two cases for a and b as short hand a/b (*e.g.* $\mathbf{Z}_{a/b}$). We substitute the unknown $\mathbf{Z}$ with $\mathbf{Z}_{a/b}$ wherever possible. *Algorithm 1* shows our direct KSFA approach. The embedding (or *testing*) of a new sample is given in *Algorithm 2*.

3.4 Incremental Slow Feature Analysis with Indefinite Kernel

In this section, we introduce our incremental SFA which holds and updates the feature representation in constant time and memory space. We base our mechanism on the mathematically exact IPCA [7].

As seen in Section 2, SFA depends on two major parts, *i.e.* the whitening of the input data and the feature optimization. In the following, we describe how we perform both steps incrementally. Fig. 1 illustrates our setup. We utilize the same notation as in Section 3.3, and indicate time steps by subscripts (*e.g.* $\mathbf{X}_t$ is the signal at time t).

[3] Although SFA [1] is originally defined in $\mathbb{R}$, its complex equivalent in $\mathbb{C}$ is found by substituting $(.)^T$ with $(.)^H$.

Algorithm 1 Batch Slow Feature Analysis with Kernel

Input: The two mapping of the training data $\mathbf{Z}_{a/b} \in \mathbb{C}^{2p \times n_{\mathbf{Z}}}$, the time derivatives $\dot{\mathbf{Z}}_{a/b} = \mathbf{Z}_{a/b}\mathbf{P}_n \in \mathbb{C}^{2p \times n_{\dot{\mathbf{Z}}}}$ and the maximum number of components k_1 for the data whitening.
Output: The data projections $\mathbf{B}_{a/b}$ with sorted components according to slowness, and the data means $\boldsymbol{\mu}_{\mathbf{Z}_{a/b}}$ and $\boldsymbol{\mu}_{\dot{\mathbf{Z}}_{a/b}}$ of the mapped signal and its derivative.

1: Compute $\boldsymbol{\mu}_{\mathbf{Z}_{a/b}} = \mathbf{Z}_{a/b}\frac{1}{n_{\mathbf{Z}}}\mathbf{1}_{n_{\mathbf{Z}} \times 1}$ and $\bar{\mathbf{Z}}_{a/b} = \mathbf{Z}_{a/b} - \boldsymbol{\mu}_{\mathbf{Z}_{a/b}}\mathbf{1}_{1 \times n_{\mathbf{Z}}}$.
2: Find $\bar{\mathbf{Z}}_a^H \bar{\mathbf{Z}}_b = \boldsymbol{\Omega}\boldsymbol{\Lambda}\boldsymbol{\Omega}^H$ and the reduced set $\boldsymbol{\Omega}_{k_1} \in \mathbb{C}^{n_{\mathbf{Z}} \times k_1}$ and $\boldsymbol{\Lambda}_{k_1} \in \mathbb{R}^{k_1 \times k_1}$ which is related to the k_1 eigenvalues with largest magnitude in $|\boldsymbol{\Lambda}|$.
3: Set $\tilde{\mathbf{W}}_{k_1} = \tilde{\mathbf{U}}_{k_1}\mathbf{D}_{k_1}^{-1} = \boldsymbol{\Omega}_{k_1}|\boldsymbol{\Lambda}_{k_1}|^{-\frac{1}{2}}|\boldsymbol{\Lambda}_{k_1}|^{-\frac{1}{2}} = \boldsymbol{\Omega}_{k_1}|\boldsymbol{\Lambda}_{k_1}|^{-1}$.
4: Calculate $\boldsymbol{\mu}_{\dot{\mathbf{Z}}_{a/b}} = \dot{\mathbf{Z}}_{a/b}\frac{1}{n_{\dot{\mathbf{Z}}}}\mathbf{1}_{n_{\dot{\mathbf{Z}}} \times 1}$ and $\dot{\bar{\mathbf{Z}}}_{a/b} = \dot{\mathbf{Z}}_{a/b} - \boldsymbol{\mu}_{\dot{\mathbf{Z}}_{a/b}}\mathbf{1}_{1 \times n_{\dot{\mathbf{Z}}}}$.
5: Compute $\tilde{\mathbf{W}}_{k_1}^H \bar{\mathbf{Z}}_a^H \dot{\bar{\mathbf{Z}}}_b \dot{\bar{\mathbf{Z}}}_a^H \bar{\mathbf{Z}}_b \tilde{\mathbf{W}}_{k_1} = \mathbf{A}\mathbf{H}\mathbf{A}^H$.
6: Reorganize $\mathbf{A}$'s components in relation to the ascending eigenvalues in $\mathbf{H}$, set $\mathbf{B}_{a/b} = \bar{\mathbf{Z}}_{a/b}\tilde{\mathbf{W}}\mathbf{A}$.

Algorithm 2 Testing with Kernel

Input: The mapping of the to-be-tested sample $\mathbf{z}_a \in \mathbb{C}^{2p}$, the data projection $\mathbf{B}_b$ with sorted components, the number k_2 of slow features to be used, and the data means $\boldsymbol{\mu}_{\mathbf{Z}a}$ and $\boldsymbol{\mu}_{\dot{\mathbf{Z}}a}$ of the mapping of the original signal and its derivative.
Output: The output signal $\mathbf{o} \in \mathbb{C}^{k_2}$.

1: Compute $\bar{\mathbf{z}}_a = \mathbf{z}_a - \boldsymbol{\mu}_{\mathbf{Z}_a} - \boldsymbol{\mu}_{\dot{\mathbf{Z}}_a}$.
2: Find $\mathbf{o} = \mathbf{B}_{b_{k_2}}^H \bar{\mathbf{z}}_a$ where $\mathbf{B}_{b_{k_2}} \in \mathbb{C}^{2p \times k_2}$ consists of the first k_2 rows of $\mathbf{B}_b$.

Incremental Whitening. Let us assume we are given an eigenspace $\mathbf{U}_{a_{t-1}/b_{t-1}}$ and $\mathbf{D}_{t-1}$ of the input data $\mathbf{Z}_{a_{t-1}/b_{t-1}}$ either from the previous time $t-1$, or computed by a batch algorithm at time 0. We want to update our whitening projections $\mathbf{W}_{a_{t-1}/b_{t-1}} = \bar{\mathbf{Z}}_{a_{t-1}/b_{t-1}}\tilde{\mathbf{W}}_{t-1}$ to additionally incorporate a new set of data samples $\mathbf{X}_\delta$, such that all the information in $\mathbf{X}_t = [\mathbf{X}_{t-1}\,\mathbf{X}_\delta]$ is represented. First, we compute the mappings $\mathbf{Z}_{a_\delta/b_\delta}$. Then we calculate the means $\boldsymbol{\mu}_{\mathbf{Z}_{a_\delta/b_\delta}}$ and center matrices $\bar{\mathbf{Z}}_{a_\delta/b_\delta}$. The data means $\boldsymbol{\mu}_{\mathbf{Z}_{a_{t-1}/b_{t-1}}}$ and the SVD of the centered data $\bar{\mathbf{Z}}_{a_{t-1}/b_{t-1}}$ are updated *via* IPCA in Krein space [14] (we assume SVD with positive and negative eigenvalues).

Let $\mathbf{X}_{t-1}$ contain $n_{\mathbf{Z}_{t-1}}$ samples, and the new data $\mathbf{X}_\delta$ consists of $n_{\mathbf{Z}_\delta}$ input vectors. Then we find the new means as

$$\boldsymbol{\mu}_{\mathbf{Z}_{a_t/b_t}} = \frac{n_{\mathbf{Z}_{t-1}}\boldsymbol{\mu}_{\mathbf{Z}_{a_{t-1}/b_{t-1}}} + n_{\mathbf{Z}_\delta}\boldsymbol{\mu}_{\mathbf{Z}_{a_\delta/b_\delta}}}{n_{\mathbf{Z}_{t-1}} + n_{\mathbf{Z}_\delta}}. \tag{19}$$

We want to compute the SVD of the new data matrix $\bar{\mathbf{Z}}_{a_t/b_t}$ to find $\mathbf{W}_{a_t/b_t}$. For the sake of simplicity, let $\boldsymbol{\mu}_{\mathbf{Z}_{a_t/b_t}} = \boldsymbol{\mu}_{\mathbf{Z}_{a_{t-1}/b_{t-1}}}$. The principle components of $\bar{\mathbf{Z}}_{a_\delta/b_\delta}$ which are not represented in $\mathbf{U}_{a_{t-1}/b_{t-1}}$ can be derived by subtracting the projected data from the original. Analogously to [10], we find the ED of

$$\left(\bar{\mathbf{Z}}_{a_\delta} - \mathbf{U}_{a_{t-1}}\mathbf{U}_{a_{t-1}}^*\bar{\mathbf{Z}}_{b_\delta}\right)^* \left(\bar{\mathbf{Z}}_{b_\delta} - \mathbf{U}_{b_{t-1}}\mathbf{U}_{a_{t-1}}^*\bar{\mathbf{Z}}_{b_\delta}\right) = \boldsymbol{\Theta}\boldsymbol{\Delta}\boldsymbol{\Theta}^H \tag{20}$$

and set $\mathbf{K}_\delta = |\boldsymbol{\Delta}^{\frac{1}{2}}|\boldsymbol{\Theta}^H$ and $\mathbf{U}_{a_\delta/b_\delta} = \left(\bar{\mathbf{Z}}_{a_\delta/b_\delta} - \mathbf{U}_{a_{t-1}/b_{t-1}}\mathbf{U}_{a_{t-1}}^*\bar{\mathbf{Z}}_{b_\delta}\right)\boldsymbol{\Theta}|\boldsymbol{\Delta}|^{-\frac{1}{2}}$.

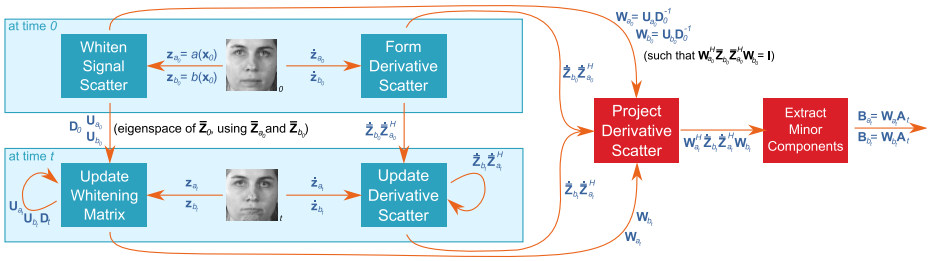

Fig. 1. Illustration of the work flow for our incremental KSFA algorithm.

Then, as shown in [14], the new SVD of the centered data is given by

$$\bar{\mathbf{Z}}_{a_t/b_t} = \left[\mathbf{U}_{a_{t-1}/b_{t-1}}\ \mathbf{U}_{a_\delta/b_\delta}\right] \begin{bmatrix} \mathbf{D}_{t-1} & \mathbf{U}_{a_{t-1}}^* \bar{\mathbf{Z}}_{b_\delta} \\ \mathbf{0} & \mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{t-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}^H \tag{21}$$

which is solved by SVD of the middle matrix, as eq. (21) becomes

$$\bar{\mathbf{Z}}_{a_t/b_t} = \left[\left[\mathbf{U}_{a_{t-1}/b_{t-1}}\ \mathbf{U}_{a_\delta/b_\delta}\right] \tilde{\mathbf{U}}\right] \left[\tilde{\mathbf{D}}\right] \left[\tilde{\mathbf{V}}^H \begin{bmatrix} \mathbf{V}_{t-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}^H\right] = \mathbf{U}_{a_t/b_t} \mathbf{D}_t \mathbf{V}_t^H \tag{22}$$

Thus, the corresponding update of the eigenspectrum is then provided by $\mathbf{U}_{a_t/b_t} = \left[\mathbf{U}_{a_{t-1}/b_{t-1}}\ \mathbf{U}_{a_\delta/b_\delta}\right] \tilde{\mathbf{U}}$ and $\mathbf{D}_t = |\tilde{\mathbf{D}}|$, and the whitening projection is computed as $\mathbf{W}_{a_t/b_t} = \mathbf{U}_{a_t/b_t} \mathbf{D}_t^{-1}$. An optional dimensionality reduction may be applied as in Section 3.3. Note, $\mathbf{V}_t$ is not required to be known or stored in memory.

In general, to allow for the case in which the mean has changed, we introduce a correcting term into the matrix of the new data, to form [8]

$$\hat{\mathbf{Z}}_{a_\delta/b_\delta} = \left[\bar{\mathbf{Z}}_{a_\delta/b_\delta} \qquad \sqrt{\frac{nm}{n+m}} \left(\boldsymbol{\mu}_{\mathbf{Z}_{a_{t-1}/b_{t-1}}} - \boldsymbol{\mu}_{\mathbf{Z}_{a_\delta/b_\delta}}\right)\right]. \tag{23}$$

Equations (20) and (21) are then computed with $\hat{\mathbf{Z}}_{a_\delta/b_\delta}$ instead of $\bar{\mathbf{Z}}_{a_\delta/b_\delta}$.

Slow Feature Update. When given a new set of $n_{\dot{\mathbf{Z}}_\delta}$ time derivatives $\dot{\mathbf{Z}}_{a_\delta/b_\delta}$, we need to update the slow features. For this, we initially propose an update of the scatter matrix $\dot{\bar{\mathbf{Z}}}_{t-1} \dot{\bar{\mathbf{Z}}}_{t-1}^T$ and the means $\boldsymbol{\mu}_{\dot{\mathbf{Z}}_{a_{t-1}/b_{t-1}}}$.

Let us assume $n_{\dot{\mathbf{Z}}_{t-1}}$ samples are represented in $\dot{\mathbf{Z}}_{a_{t-1}/b_{t-1}}$. We first find the means $\boldsymbol{\mu}_{\dot{\mathbf{Z}}_{a_\delta/b_\delta}}$ and centered data matrices $\dot{\bar{\mathbf{Z}}}_{a_\delta/b_\delta}$ of the new elements. Then, we update the scatter matrix as [8]

$$\dot{\bar{\mathbf{Z}}}_{b_t} \dot{\bar{\mathbf{Z}}}_{a_t}^H = \dot{\bar{\mathbf{Z}}}_{b_{t-1}} \dot{\bar{\mathbf{Z}}}_{a_{t-1}}^H + \dot{\bar{\mathbf{Z}}}_{b_\delta} \dot{\bar{\mathbf{Z}}}_{a_\delta}^H + \frac{nm}{n+m} \left(\boldsymbol{\mu}_{\dot{\mathbf{Z}}_{b_{t-1}}} - \boldsymbol{\mu}_{\dot{\mathbf{Z}}_{a_\delta}}\right) \left(\boldsymbol{\mu}_{\dot{\mathbf{Z}}_{a_{t-1}}} - \boldsymbol{\mu}_{\dot{\mathbf{Z}}_{b_\delta}}\right)^H \tag{24}$$

and the new means are found analogously to eq. (19). Finally, we calculate the new feature functions by an ED of $\mathbf{W}_{a_t}^H \dot{\bar{\mathbf{Z}}}_{b_t} \dot{\bar{\mathbf{Z}}}_{a_t}^H \mathbf{W}_{b_t} \in \mathbb{C}^{k_1 \times k_1}$ as in Section 3.3 before. Note, again, the unknown mapping $\phi(.)$ is not required.

Forgetting Factor. We utilize a forgetting factor f which acts as weight for old data (usually $0 \ll f < 1$) as introduced in [7, 8]. First, we adjust the update of the mean and the mean correction term in eq. (23). As in [8], we set

$$\boldsymbol{\mu}_{\mathbf{Z}_{a_t/b_t}} = \frac{f n_{\mathbf{Z}_{t-1}}}{f n_{\mathbf{Z}_{t-1}} + n_{\mathbf{Z}_\delta}} \boldsymbol{\mu}_{\mathbf{Z}_{a_{t-1}/b_{t-1}}} + \frac{n_{\mathbf{Z}_\delta}}{f n_{\mathbf{Z}_{t-1}} + n_{\mathbf{Z}_\delta}} \boldsymbol{\mu}_{\mathbf{Z}_{a_\delta/b_\delta}}. \tag{25}$$

Analogously to [8], the correction in $\hat{\mathbf{Z}}_{a_\delta/b_\delta}$ is given by[4]

$$\hat{\mathbf{Z}}_{a_\delta/b_\delta} = \left[\bar{\mathbf{Z}}_{a_\delta/b_\delta} \;\; \frac{\sqrt{f^2 n_{\mathbf{Z}_\delta} n_{\mathbf{Z}_{t-1}} \left(n_{\mathbf{Z}_\delta} + n_{\mathbf{Z}_{t-1}}\right)}}{f n_{\mathbf{Z}_{t-1}} + n_{\mathbf{Z}_\delta}} \left(\boldsymbol{\mu}_{\mathbf{Z}_{a_{t-1}/b_{t-1}}} - \boldsymbol{\mu}_{\mathbf{Z}_{a_\delta/b_\delta}} \right) \right] \tag{26}$$

The update of the whitening is then computed with $\hat{\mathbf{Z}}_{a_\delta/b_\delta}$. Note, similar to [8], we apply f to the previous eigenvalues, *i.e.* we set $\mathbf{D}_{t-1} = f\mathbf{D}_{t-1}$. Finally, we incorporate f in the total number of samples $n_{\mathbf{Z}_t} = f n_{\mathbf{Z}_{t-1}} + n_{\mathbf{Z}_\delta}$.

To introduce a forgetting factor into the update of the scatter matrix in eq. (24) a similarly modification is required. Analogously to [8] we find

$$\begin{aligned} \dot{\hat{\mathbf{Z}}}_{b_t} \dot{\hat{\mathbf{Z}}}_{a_t}^H = {} & \frac{f^2 n_{\mathbf{Z}_\delta} n_{\mathbf{Z}_{t-1}} \left(n_{\mathbf{Z}_\delta} + n_{\mathbf{Z}_{t-1}}\right)}{\left(f n_{\mathbf{Z}_{t-1}} + n_{\mathbf{Z}_\delta}\right)^2} \left(\boldsymbol{\mu}_{\dot{\mathbf{Z}}_{b_{t-1}}} - \boldsymbol{\mu}_{\dot{\mathbf{Z}}_{b_\delta}} \right) \left(\boldsymbol{\mu}_{\dot{\mathbf{Z}}_{a_{t-1}}} - \boldsymbol{\mu}_{\dot{\mathbf{Z}}_{b_\delta}} \right)^H \\ & + f^2 \dot{\hat{\mathbf{Z}}}_{b_{t-1}} \dot{\hat{\mathbf{Z}}}_{a_{t-1}}^H + \dot{\hat{\mathbf{Z}}}_{b_\delta} \dot{\hat{\mathbf{Z}}}_{a_\delta}^H \end{aligned} \tag{27}$$

Finally, the number of elements is adjusted to $n_{\dot{\mathbf{Z}}_t} = f n_{\dot{\mathbf{Z}}_{t-1}} + n_{\dot{\mathbf{Z}}_\delta}$.

4 Real-time Temporal Video Segmentation

In this section, we employ our incremental KSFA with indefinite kernel to the problem of temporal video segmentation. The segmentation of a video sequence in time is closely related to finding consecutive frames which have large differences in their slow features [2].

SFA minimizes the slowness of a signal, eq. (1), which is the squared sum of Euclidian distances of the slow features between consecutive samples of the complete signal. Analogously, we define the *change* of signal $\mathbf{z}_i$, after time t, as the squared Euclidian difference in slow features towards its previous signal

$$c_{k_t}(\mathbf{z}_i) = \mathbf{B}_{k_t}^H (\mathbf{z}_i - \mathbf{z}_{i-1})(\mathbf{z}_i - \mathbf{z}_{i-1})^H \mathbf{B}_{k_t} \tag{28}$$

where $c_{k_t}(.)$ depends on the number of utilized slow features k. When a new activity has started, the change is expected to be unusually large. To compare the current change $c_{k_{t-1}}(\mathbf{z}_t)$ to previous data we utilize the average of all $c_{k_{t-1}}(\mathbf{z}_i)$, $i = 1, \ldots, t-1$. However, a trivial update of the mean is not possible, as $c_{k_{t-1}}(.)$ changes with each time interval. We want to compute the average change of previous time-steps without keeping the whole signal in memory.

[4] The correction with f is not in [8], but is trivial to find, following their derivation.

Considering our update, for eq. (18), we found the ED of $\mathbf{W}_{t-1}^{*}\dot{\tilde{\mathbf{Z}}}_{t-1}\dot{\tilde{\mathbf{Z}}}_{t-1}^{*}\mathbf{W}_{t-1}$ as $\mathbf{A}_{t-1}\mathbf{H}_{t-1}\mathbf{A}_{t-1}^{H}$. As $\dot{\tilde{\mathbf{Z}}}_{t-1} = [\mathbf{z}_2 \cdots \mathbf{z}_{t-1}] - [\mathbf{z}_1 \cdots \mathbf{z}_{t-2}] - \boldsymbol{\mu}_{\dot{\mathbf{Z}}_{t-1}}\mathbf{1}_{1\times n_{\dot{\mathbf{z}}_{t-1}}}$, the sum of the k largest eigenvalues in $\mathbf{H}_{t-1}$ is nearly equivalent to $\sum_{1}^{t-1} c_{k_{t-1}}(\mathbf{z}_i)$. The difference is caused by $\boldsymbol{\mu}_{\dot{\mathbf{Z}}_{t-1}}$, and thus we compute the average as

$$\mu_{k_{t-1}} = \frac{tr\left(\mathbf{H}_{k_{t-1}}\right)}{n_{\dot{\mathbf{Z}}_{t-1}}} + \boldsymbol{\mu}_{\dot{\mathbf{Z}}_{t-1}}^{*}\mathbf{W}_{t-1}\mathbf{A}_{k_{t-1}}\mathbf{A}_{k_{t-1}}^{H}\mathbf{W}_{t-1}^{*}\boldsymbol{\mu}_{\dot{\mathbf{Z}}_{t-1}}. \tag{29}$$

We utilize the ratio between $c_{k_{t-1}}(\mathbf{z}_t)$ and $\mu_{k_{t-1}}$ to judge how significant the change at the current time-step is. The frames with unusually large variations in slow features (according to a threshold), are then used to segment the different parts of the video stream. For each time-step, we first analyze the significance of variation, and then update the SFA with the new data.

An optional median filter of size n ($n = 8$ for our system) may be applied to smoothen the change detection. Although, this introduces a delay of $\frac{n}{2}$ frames, as the results of the surrounding data is needed, we found it to be beneficial, as outliers are suppressed. If immediate output is required, we skip this part.

5 Evaluation

In this section, we evaluate our system in a number of setups. First, we provide a proof of concept, which shows that our incremental SFA can be used for identifying change. Second, we compare the utilization of our kernel quantitatively to SFA with linear and quadratic features, using samples of the MMI Facial Expression Database (MMI) [19] for the detection of onset and offset. Finally, we test qualitatively on videos from YouTube and the Ballet data set in [20].

5.1 Proof of Concept

We test our incremental SFA on the input signal given by $\mathbf{x}_i = [\sin(m_i) + \cos(11m_i)^2, \cos(11m_i)]^T$ (Fig. 2). A quadratic expansion is used ($a(\mathbf{x}_i) = b(\mathbf{x}_i) = \left[\mathbf{x}_i(1), \mathbf{x}_i(2), \mathbf{x}_i(1)\mathbf{x}_i(2), \mathbf{x}_i(1)^2, \mathbf{x}_i(2)^2\right]^T$). We use 500 equally distributed samples in the range $[0, 4\pi]$. Add each time-step, we feed another input vector into our system. The estimated change ratio of each sample, based on 1 slow feature, is computed as soon as it became available (no filter is applied). The ground truth is the change ratio of the whole sequence when known *a priori*.

Fig. 2 visualizes the results. With more data, the incremental SFA becomes increasingly accurate. At first, the trend of the signal is unknown, and thus most variations in the features are large. As the incremental SFA learns the behavior of the input sequence, it can better estimate the significance of the change.

5.2 Quantitative Evaluation

In our second experiment, we compare incremental SFA using our input features (K-SFA), given by the mappings in eq. (10), with the quadratic expansion (Q-SFA) and linear features (L-SFA) (when $a(\mathbf{x}_i) = b(\mathbf{x}_i) = \mathbf{x}_i$). The MMI data set

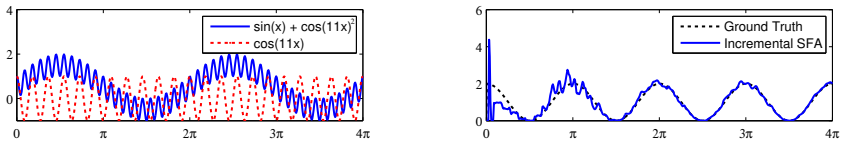

Fig. 2. The results of our incremental change detection algorithm (right) compared to ground truth for which the input signal (left) is known completely *a priori*.

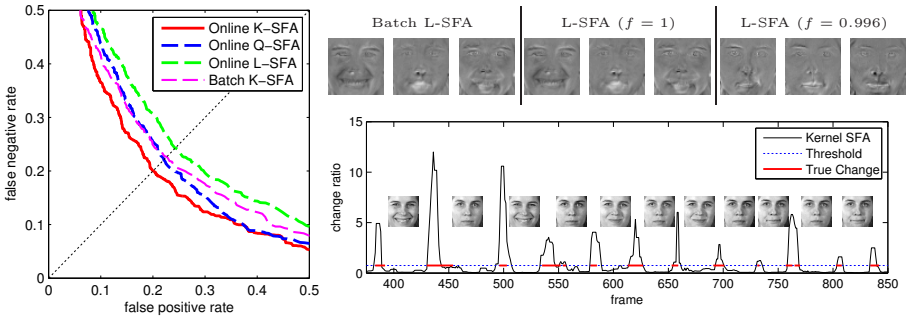

Fig. 3. The false positive over false negative rate of our Online K-SFA in comparison to other versions (left), the 3 slowest projections for L-SFA after 900 frames (top-right), and the output of our online change detection with Online K-SFA (bottom-right).

[19] is used. We utilize the first 60 expressions of the first subject. Each action in MMI is labeled by onset, apex and offset. The onset and offset indicate the start and end of the expression respectively. We utilize these labels as ground truth as they mark the frames in which the activity in the video changes.

Initially we concatenate all videos and employ our tracker in [14] to extract aligned images (50×50 pixel) of several activities. Then we optimize both setups with respect to the number of components used for the whitening k_1, the forgetting factor f and the number of slow features k_2. All methods perform best with $f = 0.996$ (≈ 250 frames), and $k_2 = 3$. However, K-SFA performed best for $k_1 = 10$, which is much less than needed for L-SFA, which required $k_1 = 80$, and Q-SFA for which $k_1 = 20$. Additionally, we include the batch version of K-SFA (Batch K-SFA), for which we compute the change ratio with all samples known *a priori*. We plot the false positive over the false negative rate (Fig. 3). The equal error rate is best for Online K-SFA. Batch K-SFA is inherently non-adaptive and, thus, performs worse. The spectrums of Online K-SFA is given in Fig. 3.

Fig. 3 also shows the top 3 projections after frame 900 for L-SKA[5] as (i) batch setup, (ii) with $f = 1$ (no forgetting), and (iii) with $f = 0.996$. We compare (i) with (ii). The resulting projections are virtually equivalent, which validates our update procedure. Notice, how the slowest projection relates to the smile around

[5] L-SFA is chosen to aid visualization, as the projections remain in the original space.

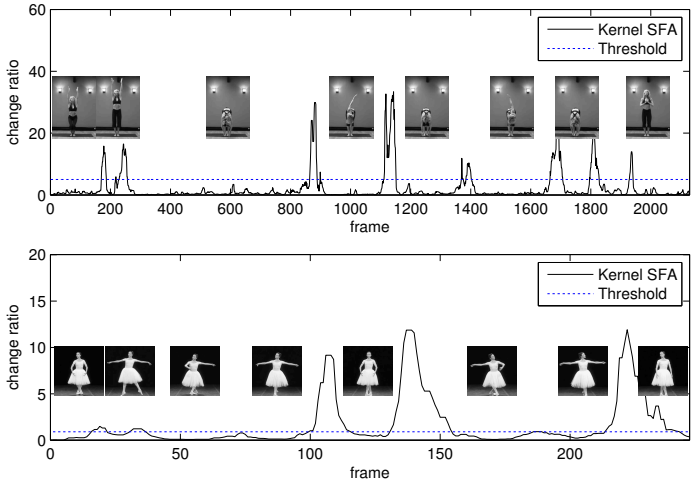

Fig. 4. Frames of the segmentation for yoga (top) and ballet (bottom) scenes.

frame 530. For (iii) the effect of f becomes apparent, as the projections for this setup are most relevant to later expressions. Here, the smile is no longer visible.

Finally, we emphasize the advantage of our domain-specific kernel in Krein space. Although our kernel utilizes a small reduced set ($k_1 = 10$), it yields the best performance. The second best online approach is Q-SFA which needs $k_1 = 20$ components for the whitening. Note, if fewer components are used eq. (21) is faster to compute. Furthermore, $a(.)$ and $b(.)$ can be computed in linear time and memory. The quadratic expansion is polynomial.

5.3 Qualitative Evaluation

We conclude our experiments with a selection of video sequences from different scenarios. Fig. 4 shows the extracted spectrums of a yoga sequence[6] and an example of segmented ballet videos [20]. We use the same parameters as in the previous section. The temporal video segments are clearly visible. Please visit `http://www.doc.ic.ac.uk/~sl609/sfa/` for videos and source code.

6 Conclusion

We utilize a domain-specific robust indefinite kernel for measuring visual similarity. We then developed slow feature analysis for our indefinite kernel in Krein space. Additionally, we proposed a direct incremental KSFA which does not rely on a reduced set, as we utilizes the special two mappings which equal our kernel. Finally, we employ our learning framework in SFA's first online temporal video segmentation algorithm, and perform qualitative and quantitative evaluation.

[6] Taken from `http://www.youtube.com/watch?v=ziVctQnyvwE`

Acknowledgement. The work presented in this paper has been funded by the European Research Council under the ERC Starting Grant agreement no. ERC-2007-StG-203143 (MAHNOB). S. Liwicki is funded by an EPSRC Studentship.

References

1. Wiskott, L., Sejnowski, T.: Slow Feature Analysis: Unsupervised Learning of Invariances. Neural Computation **14** (2002) 715 – 770
2. Nater, A., Grabner, H., Van Gool, L.: Temporal Relations in Videos for Unsupervised Activity Analysis. In: Mach. Learning. (2004) 78–86
3. Kompella, V., Luciw, M., Schmidhuber, J.: Incremental Slow Feature Analysis. In: IJCAI'11. (2011) 1354 – 1359
4. Zhang, Z., Tao, D.: Slow Feature Analysis for Human Action Recognition. IEEE Trans. Pattern Anal. Mach. Intell. **34** (2012) 436 – 450
5. Böhmer, W., Grünewälder, S., Nickisch, H., Obermayer, K.: Regularized Sparse Kernel Slow Feature Analysis. In: ECML'11. (2011) 235 – 248
6. Franzius, M., Sprekeler, H., Wiskott, L.: Slowness and Sparseness Lead to Place, Head-Direction, and Spatial-View Cells. PLoS Comput. Biol. **3** (2007) 1605 – 1622
7. Levy, A., Lindenbaum, M.: Sequential Karhunen-Loeve Basis Extraction and its Application to Images. IEEE Trans. Image Process. **9** (2000) 1371 – 1374
8. Ross, D., Lim, J., Lin, R., Yang, M.: Incremental Learning for Robust Visual Tracking. Int. Journal of Comp. Vision **77** (2008) 125 – 141
9. Weng, J., Zhang, Y., Hwang, W.: Candid Covariance-Free Incremental Principal Component Analysis. IEEE Trans. Pattern Anal. Mach. Intell. **25** (2003) 1034 – 1040
10. Chin, T., Suter, D.: Incremental Kernel Principal Component Analysis. IEEE Trans. Image Process. **16** (2007) 1662 – 1674
11. Zhou, F., De la Torre, F., Cohn, J.: Unsupervised Discovery of Facial Events. In: CVPR'10. (2010) 2574 – 2581
12. Turaga, P., Veeraraghavan, A., Chellappa, R.: Unsupervised View and Rate Invariant Clustering of Videosequences. Comp. Vision and Image Understanding **113** (2009) 353 – 371
13. Hoai, M., Lan, Z., De la Torre, F.: Joint Segmentation and Classification of Human Actions in Video. In: CVPR'11. (2011) 3265 3272
14. Liwicki, S., Zafeiriou, S., Tzimiropoulos, G., Pantic, M.: Efficient Online Subspace Learning with an Indefinite Kernel for Visual Tracking and Recognition. IEEE Trans. Neu. Net. Learn. Systems **23** (2012) 1624–1636
15. Tzimiropoulos, G., Argyriou, V., Zafeiriou, S., Stathaki, T.: Robust FFT-Based Scale-Invariant Image Registration with Image Gradients. IEEE Trans. Pattern Anal. Mach. Intell. **32** (2010) 1899 – 1906
16. Turk, M., Pentland, A.: Eigenfaces for Recognition. Journal of Cognitive Neuroscience **3** (1991) 71 – 86
17. Pękalska, E., Haasdonk, B.: Kernel Discriminant Analysis for Positive Definite and Indefinite Kernels. IEEE Trans. Pattern Anal. Mach. Intell. **31** (2009) 1017 – 1032
18. Hassibi, B., Sayed, A., Kailath, T.: Linear Estimation in Krein Spaces. I. Theory. IEEE Trans. Automatic Control **41** (1996) 18 –33
19. Valstar, M., Pantic, M.: Induced Disgust, Happiness and Surprise: an Addition to the MMI Facial Expression Database. In: LREC'10. (2010) 65 – 70
20. Fathi, A., Mori, G.: Action Recognition by Learning Mid-level Motion Features. In: CVPR'08. (2008)