Efficient Online Subspace Learning With an Indefinite Kernel for Visual Tracking and Recognition

Stephan Liwicki, *Student Member, IEEE*, Stefanos Zafeiriou, *Member, IEEE*, Georgios Tzimiropoulos, *Member, IEEE*, and Maja Pantic, *Fellow, IEEE*

***Abstract*— We propose an exact framework for online learning with a family of indefinite (not positive) kernels. As we study the case of nonpositive kernels, we first show how to extend kernel principal component analysis (KPCA) from a reproducing kernel Hilbert space to Krein space. We then formulate an incremental KPCA in Krein space that does not require the calculation of preimages and therefore is both efficient and exact. Our approach has been motivated by the application of visual tracking for which we wish to employ a robust gradient-based kernel. We use the proposed nonlinear appearance model learned online via KPCA in Krein space for visual tracking in many popular and difficult tracking scenarios. We also show applications of our kernel framework for the problem of face recognition.**

***Index Terms*— Gradient-based kernel, online kernel learning, principal component analysis with indefinite kernels, recognition, robust tracking.**

I. Introduction

With ever-increasing importance of realizing online and real-time applications, incremental learning methods have become a popular research topic. Many online learning methods have been recently proposed. For example, in [1], an online algorithm to learn from stream data was introduced. A framework for learning with nonstationary environments, for which the classes and data distribution change over time, was presented in [2]. Motivated by the cerebral cortex, [3] proposed an online methodology for classification and regression. Online blind source separation was introduced in [4], and several variations of incremental principal component analysis (PCA) have also been recently introduced [5]–[8].

In this paper, we particularly focus on online and incremental kernel learning. Online kernel learning for classification, regression, novelty and change detection, subspace learning, and feature extraction is a very active research field [9]–[27]. Typically, the online classification or regression function is written as a weighted sum of kernel combination of samples from a set of stored instances, usually referred to as a "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.

One of the major challenges in online learning is that the support set may grow to become arbitrarily large over time [9]–[27]. In [12], online kernel algorithms for classification, regression, and novelty detection are proposed, based on a stochastic gradient descent algorithm in Hilbert space. In order to avoid the arbitrary growth of the support set, the authors adopt simple truncation and shrinking strategies. In [13], an online kernel regression algorithm is proposed based on constructing and solving minimum mean-squared-error optimization problems with Mercer kernels. In order to regularize solutions and keep the complexity of the algorithm bound, a sequential sparsification process is adopted. In [25], the so-called Projectron algorithm is proposed, which neither truncates nor discards instances. In order to keep the support set bound, the algorithm projects the samples onto the space spanned by the support set. If this is impossible without greater loss, the samples are added to the support set. It is proven that, by following the Projectron algorithm, the support set and, therefore, the online hypothesis are guaranteed to converge. The drawback is that the size of support cannot be predicted in advance. To cope with this, a parameter that provides a tradeoff between accuracy and sparsity is introduced. In [26], online regression algorithms are proposed that use an alternative model-reduction criterion. Instead of using sparsification procedures, the increase in the number of variables is controlled by a coherence parameter, a fundamental quantity that characterizes the behavior of dictionaries in sparse approximation problems.

Although many methods have been proposed for online kernel learning for classification and regression, limited research has been conducted for online subspace learning with kernels.

Manuscript received May 1, 2011; revised April 4, 2012; accepted June 16, 2012. Date of publication August 15, 2012; date of current version September 10, 2012. This work was supported in part by the European Research Council (ERC) under the ERC Starting Grant Agreement ERC-2007-StG-203143. The work of S. Liwicki was supported by the Engineering and Physical Science Research Council DTA Studentship. The work of G. Tzimiropoulos was supported in part by the European Community's 7th Framework Programme FP7/2007-2013 under Grant Agreement 288235 (FROG).

S. Liwicki and S. Zafeiriou are with the Department of Computing, Imperial College London, London SW7 2AZ, U.K. (e-mail: sl609@imperial.ac.uk; s.zafeiriou@imperial.ac.uk).

G. Tzimiropoulos is with the School of Computer Science, University of Lincoln, Lincoln LN6 7TS, U.K., and also with the Department of Computing, Imperial College London, London SW7 2AZ, U.K. (e-mail: gtzimiropoulos@lincoln.ac.uk).

M. Pantic is with the Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede-Noord 7522NB, The Netherlands, and also with the Department of Computing, Imperial College London, London SW7 2AZ, U.K. (e-mail: m.pantic@imperial.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2012.2208654

This research has mainly revolved around the development of incremental kernel PCA (KPCA) [16], [21], [22] and kernel singular value decomposition [28] algorithms. One of the first incremental KPCA algorithms with reproducing kernel Hilbert space (RKHS) is proposed in [16]. This algorithm is essentially the kernelization of the generalized Hebbian algorithm, which has operational characteristics similar to those of a single-layer feedforward neural network. Gain adaptation methods that improve convergence of the kernel Hebbian algorithm are proposed in [22]. An incremental KPCA algorithm with Hilbert spaces, which kernelizes an exact algorithm for incremental PCA [5], [6], is proposed in [21]. In this method, in order to maintain constant update speed, the authors construct reduced set expansions, by means of preimages, of the kernel principal components and the mean. The main drawbacks of this method are: 1) the reduced set representation provides only an approximation to the exact solution and 2) the proposed optimization problem for finding the expansions inevitably increases the complexity of the algorithm.

In this paper, we propose an exact framework for online learning with a family of indefinite (nonpositive) kernels. As we study the case of nonpositive kernels, we first show how to extend KPCA from an RKHS to Krein space. Note that all the above-described online kernel subspace learning algorithms support only arbitrarily chosen positive definite kernels [e.g., polynomial or Gaussian radial basis function (RBF)]. We then propose a kernel that allows the formulation of an incremental KPCA in Krein space which does not require the calculation of preimages and therefore is both efficient and exact.

Our approach has been motivated by the application of visual tracking. In fact, many subspace learning algorithms have been either developed or evaluated for the application of visual tracking [6], [21], [29]–[32]. Visual trackers aim to locate a predefined target object in a video sequence. Typically, a tracker consists of three main components.

1) The image representation defines the low-level features that are extracted from the frames of the video sequence. Widely used features include raw pixel intensities [6], [21], [33], color [34], gradient [30], [35], Haar-like features [36], and local binary patterns [37].
2) The appearance model stands usually for a statistical model of the target. This is where incremental subspace learning algorithms are usually applied.
3) The motion model describes the set of parameters that define the motion of the target and its dynamics. A typical choice for the motion model is an affine or a similarity transform used in a particle filter framework [6], [33], [38].

As our experiments have shown, visual tracking using off-the-shelf kernels (such as the Gaussian RBF), which do not incorporate any problem-specific prior knowledge, results in loss of robustness and accuracy. To tackle this problem, we employ an indefinite robust gradient-based kernel, inspired by recently proposed schemes for the robust estimation of large translational displacements [39]. We evaluate the performance of the proposed methodology in many popular difficult tracking scenarios, and show its applicability for face recognition and that it outperforms KPCA with a Gaussian RBF kernel and standard ℓ_2-norm PCA.

In summary, our contributions are as follows.

1) We design a robust indefinite (nonpositive) kernel for measuring visual similarity.
2) We formulate KPCA in Krein space.
3) We propose an accurate incremental KPCA in Krein space, which exploits the properties of our kernel and does not require a reduced set representation.
4) We apply our learning framework to the application of visual tracking and achieve state-of-the-art performance.
5) We also apply our KPCA to face recognition, where we show better class separation properties in comparison to KPCA in Hilbert space with a Gaussian RBF kernel and standard ℓ_2-norm PCA.

In [40], a proposal is made that is closely related to the one proposed here. We would like to highlight that [40] proposes two-class classifiers based on a quadratic discriminant function in both Hilbert and Krein spaces. Our paper takes a different direction. That is, we propose subspace learning algorithms in Krein spaces for feature extraction and object representation.

The rest of this paper is organized as follows. We summarize the theory of Krein spaces and introduce our kernel in Section II. In Section III, we propose KPCA in Krein space and present our direct incremental update of the nonlinear subspace which exploits the special properties of our kernel. The visual tracker introduced in Sections IV and V presents our experimental results. Section VI concludes this paper. The interested reader is advised to visit http://www.doc.ic.ac.uk/~sl609/dikt/ for additional video results and sample code.

II. Krein Spaces and the Proposed Indefinite Kernel

Krein spaces provide feature-space representations of dissimilarities and insights on the geometry of classifiers defined with nonpositive kernels [40], [41]. An abstract space $\mathcal{K}$ is a Krein space over reals $\mathbb{R}$ if there exists an (indefinite) inner product $\langle .,. \rangle_{\mathcal{K}}: \mathcal{K} \times \mathcal{K} \rightarrow \mathbb{R}$ with the following properties [42]:

$$\begin{aligned} \langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{K}} &= \langle \mathbf{y}, \mathbf{x} \rangle_{\mathcal{K}} \\ \langle c_1 \mathbf{x} + c_2 \mathbf{z}, \mathbf{y} \rangle_{\mathcal{K}} &= c_1 \langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{K}} + c_2 \langle \mathbf{z}, \mathbf{y} \rangle_{\mathcal{K}} \end{aligned} \tag{1}$$

for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{K}$ and $c_1, c_2 \in \mathbb{R}$. $\mathcal{K}$ is composed of two vector spaces, such that $\mathcal{K} = \mathcal{K}_+ \oplus \mathcal{K}_-$. $\mathcal{K}_+$ and $\mathcal{K}_-$ describe two Hilbert spaces over $\mathbb{R}$. We denote their corresponding positive definite inner products as $\langle .,. \rangle_{\mathcal{K}_+}$ and $\langle .,. \rangle_{\mathcal{K}_-}$, respectively. The decomposition of $\mathcal{K}$ into two such 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 $\mathbf{x} = \mathbf{P}_+\mathbf{x} + \mathbf{P}_-\mathbf{x}$. The identity matrix in $\mathcal{K}$ is given by $\mathbf{I}_{\mathcal{K}} = \mathbf{P}_+ + \mathbf{P}_-$.

Let us denote by $\mathbf{x}_+ \in \mathcal{K}_+$ and $\mathbf{x}_- \in \mathcal{K}_-$, 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 nonzero vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathcal{K}$. Therefore, $\mathcal{K}_+$ is a positive subspace, while $\mathcal{K}_-$ is a negative

subspace. The inner product of $\mathcal{K}$ is defined as the difference of $\langle .,. \rangle_{\mathcal{K}_+}$ and $\langle .,. \rangle_{\mathcal{K}_-}$, i.e., for all $\mathbf{x}, \mathbf{y} \in \mathcal{K}$

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{K}} = \langle \mathbf{x}_+, \mathbf{y}_+ \rangle_{\mathcal{K}_+} - \langle \mathbf{x}_-, \mathbf{y}_- \rangle_{\mathcal{K}_-}. \tag{2}$$

A Krein space $\mathcal{K}$ 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 the basic properties of a Krein space. Its connection to the original Krein space can be written in terms of a "conjugate" by using (2) and $\mathbf{J}$, as

$$\mathbf{x}^* \mathbf{y} \triangleq \langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{K}} = \mathbf{x}^T \mathbf{J} \mathbf{y} = \langle \mathbf{J}\mathbf{x}, \mathbf{y} \rangle_{|\mathcal{K}|}. \tag{3}$$

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

In the following, we are particularly interested in finite-dimensional Krein spaces where $\mathcal{K}_+$ is isomorphic to $\mathbb{R}^p$ and $\mathcal{K}_-$ is isomorphic to $\mathbb{R}^q$. Such a Krein space describes a pseudo-Euclidean space and is characterized by its so-called signature $(p, q) \in \mathbb{N}^2$, which indicates the dimensionality p and q of the positive and negative subspaces, respectively [40]. In particular, the fundamental symmetry is given by

$$\mathbf{J} = \begin{bmatrix} \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_q \end{bmatrix} \tag{4}$$

where $\mathbf{I}_z$ is the identity matrix in $\mathbb{R}^{z \times z}$ and $\mathbf{0}$ implies zero padding. In the following, we will define an indefinite kernel with special properties that allows efficient incremental subspace learning techniques.

A. Robust Indefinite Gradient-Based Kernel for Tracking

Off-the-shelf kernels (such as the Gaussian RBF), which do not incorporate any problem-specific prior knowledge to the domain of visual tracking, often result in loss of robustness and accuracy. To tackle this deficiency, we employ an indefinite robust gradient-based kernel inspired by recently proposed schemes for the robust estimation of large translational displacements [39]. More specifically, 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 $\mathbf{G}_i = \mathbf{F}_x \star \mathbf{I}_i + j\mathbf{F}_y \star \mathbf{I}_i$, 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 $\mathbf{x}_i \in \mathbb{C}^d$ $(d = mn)$ be the d-dimensional vector obtained by writing $\mathbf{G}_i$ in lexicographical order. The gradient correlation coefficient is given by

$$\begin{aligned} s(\mathbf{x}_i, \mathbf{x}_j) &= \mathbb{R}\left\{ \mathbf{x}_i^H \mathbf{x}_j \right\} \\ &= \sum_{c=1}^{d} \mathbf{R}_i(c) \mathbf{R}_j(c) \cos\left(\Delta\boldsymbol{\theta}(c)\right) \end{aligned} \tag{5}$$

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 [39]. We propose to use a modification of this correlation as a new kernel. In particular, as the gradient magnitudes are more sensitive to outliers [39], it is very likely that the products $\mathbf{R}_i(c)\mathbf{R}_j(c)$ will be affected most. One way to circumvent this problem may be to remove the gradient magnitude from (5) as in the learning framework of [43] or the alignment framework of [44], but this could result in loss of useful information. In this paper, we split the correlation in (5) into two terms to reduce the effect of outliers to some extent. That is, we set $\mathbf{R}_i(c) = 1$ for one term and $\mathbf{R}_j(c) = 1$ for the other

$$s(\mathbf{x}_i, \mathbf{x}_j) = \sum_{c=1}^{d} \mathbf{R}_i(c) \cos\left(\Delta\boldsymbol{\theta}(c)\right) + \sum_{c=1}^{d} \mathbf{R}_j(c) \cos\left(\Delta\boldsymbol{\theta}(c)\right). \tag{6}$$

Finally, we define our kernel as the normalized version of the above correlation (see Appendix for details)

$$k(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{c=1}^{d} \mathbf{R}_i(c) \cos\left(\Delta\boldsymbol{\theta}(c)\right)}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_i^2(c) d}} + \frac{\sum_{c=1}^{d} \mathbf{R}_j(c) \cos\left(\Delta\boldsymbol{\theta}(c)\right)}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_j^2(c) d}}. \tag{7}$$

The robust properties of the proposed kernel derives from: 1) the use of gradient orientation features; 2) the way we split the magnitude; and 3) from the use of the cosine on the difference of gradient orientations (the interested reader may refer to [39], [44], and [43]).

After simple manipulations, we can write our kernel as

$$\begin{bmatrix} \frac{\mathbf{R}_i \cos(\boldsymbol{\theta}_i)}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_i^2(c) d}} \\ \frac{\mathbf{R}_i \sin(\boldsymbol{\theta}_i)}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_i^2(c) d}} \\ \cos(\boldsymbol{\theta}_i) \\ \sin(\boldsymbol{\theta}_i) \end{bmatrix}^T \begin{bmatrix} \cos(\boldsymbol{\theta}_j) \\ \sin(\boldsymbol{\theta}_j) \\ \frac{\mathbf{R}_j \cos(\boldsymbol{\theta}_j)}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_j^2(c) d}} \\ \frac{\mathbf{R}_j \sin(\boldsymbol{\theta}_j)}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_j^2(c) d}} \end{bmatrix} \tag{8}$$

where $\cos(\boldsymbol{\theta}_i) = [\cos(\boldsymbol{\theta}_i(1)) \cdots \cos(\boldsymbol{\theta}_i(d))]^T$ and $\sin(\boldsymbol{\theta}_i) = [\sin(\boldsymbol{\theta}_i(1)) \cdots \sin(\boldsymbol{\theta}_i(d))]^T$. We define two explicit mappings $a : \mathbb{C}^d \to \mathbb{R}^{4d}$ and $b : \mathbb{C}^d \to \mathbb{R}^{4d}$

$$a(\mathbf{x}_i) = \begin{bmatrix} \frac{\mathbf{R}_i \cos(\boldsymbol{\theta}_i)}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_i^2(c) d}} \\ \frac{\mathbf{R}_i \sin(\boldsymbol{\theta}_i)}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_i^2(c) d}} \\ \cos(\boldsymbol{\theta}_i) \\ \sin(\boldsymbol{\theta}_i) \end{bmatrix} \quad b(\mathbf{x}_i) = \begin{bmatrix} \cos(\boldsymbol{\theta}_i) \\ \sin(\boldsymbol{\theta}_i) \\ \frac{\mathbf{R}_i \cos(\boldsymbol{\theta}_i)}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_i^2(c) d}} \\ \frac{\mathbf{R}_i \sin(\boldsymbol{\theta}_i)}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_i^2(c) d}} \end{bmatrix}. \tag{9}$$

Given $a(.)$ and $b(.)$, our kernel $k(.,.)$ can be expressed as an example of the following family of kernels:

$$k(\mathbf{x}_i, \mathbf{x}_j) = a(\mathbf{x}_i)^T b(\mathbf{x}_j) = a(\mathbf{x}_j)^T b(\mathbf{x}_i). \tag{10}$$

When $a(.) \neq b(.)$, kernels of the form (10), which also satisfy $k(\mathbf{x}_i, \mathbf{x}_i) \geq 0$, are in general nonpositive definite, as the triangular inequality may not hold. There could exist two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ such that $k(\mathbf{x}_i, \mathbf{x}_j) > \sqrt{k(\mathbf{x}_i, \mathbf{x}_i)}\sqrt{k(\mathbf{x}_j, \mathbf{x}_j)}$. As for the proposed kernel $k(.,.)$ in (7), $k(\mathbf{x}_i, \mathbf{x}_i) = a(\mathbf{x}_i)^T b(\mathbf{x}_i) \geq 0$ it is indefinite, and defines an implicit mapping $\psi : \mathbb{C}^d \to \mathcal{K}$ into a finite-dimensional Krein space. Analogous to the Hilbert space, our kernel is equivalent to the dot product in feature space, i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle_{\mathcal{K}}$. The squared distance

in feature space is given by

$$l^2(\mathbf{x}_i, \mathbf{x}_j) = (\psi(\mathbf{x}_i) - \psi(\mathbf{x}_j))^*(\psi(\mathbf{x}_i) - \psi(\mathbf{x}_j)) = k(\mathbf{x}_i, \mathbf{x}_i) - 2k(\mathbf{x}_i, \mathbf{x}_j) + k(\mathbf{x}_j, \mathbf{x}_j). \quad (11)$$

Finally, it can be shown that $l^2(\mathbf{x}_i, \mathbf{x}_j) \geq 0$ (see Appendix).

Note that the complexity of computing the kernel remains of the computation of the kernel in $\mathcal{O}(d)$, as we extent the d-dimensional samples by a constant factor of 4 for each mapping. Finally, we emphasize that the proposed kernel is nonpositive definite. Consequently, we cannot define an implicit Hilbert feature space. In this case, the appropriate vector space where the kernel represents a dot product is a Krein space [40].

III. Direct Incremental KPCA in Krein Space

In this section we present our direct incremental KPCA in Krein space which is specifically designed to make use of the special properties of our kernel. First, we develop KPCA with nonpositive definite kernels in Krein space. We then exploit the special form of our kernel and present a direct version of KPCA. Finally, we propose our direct incremental KPCA.

A. KPCA in Krein Space

Let $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_N] \in \mathbb{C}^{d \times N}$ be a set of given samples and $\mathbf{X}_\psi = [\psi(\mathbf{x}_1) \cdots \psi(\mathbf{x}_N)]$ be their implicit mapping. Motivated by KPCA and pseudo-Euclidean embedding [40], [41], we formulate KPCA with Krein spaces.

Let us define the mean ψ_μ and the centralized matrix $\tilde{\mathbf{X}}_\psi$ as

$$\psi_\mu = \tfrac{1}{N}\mathbf{X}_\psi \mathbf{1}_N \qquad \tilde{\mathbf{X}}_\psi = \mathbf{X}_\psi \mathbf{M} \quad (12)$$

where $\mathbf{M} \triangleq \mathbf{I}_N - (1/N)\mathbf{1}_N\mathbf{1}_N^T$ and $\mathbf{1}_N$ is an N-dimensional vector containing only ones [40]. We then define the total scatter matrix in $\mathcal{K}$ as

$$\mathbf{S}_\mathcal{K} \triangleq \frac{1}{N}\sum_{i=1}^{N}(\psi(\mathbf{x}_i) - \psi_\mu)(\psi(\mathbf{x}_i) - \psi_\mu)^* = \frac{1}{N}\tilde{\mathbf{X}}_\psi\tilde{\mathbf{X}}_\psi^* = \frac{1}{N}\tilde{\mathbf{X}}_\psi\tilde{\mathbf{X}}_\psi^T\mathbf{J} = \mathbf{S}_{|\mathcal{K}|}\mathbf{J} \quad (13)$$

where $\mathbf{S}_{|\mathcal{K}|}$ is the total scatter matrix in the associated Hilbert space $|\mathcal{K}|$.

Analogous to KPCA in Hilbert space, we generalize KPCA in Krein space as follows. We wish to compute a set of projections $\mathbf{U}_o = [\mathbf{u}_1, \ldots, \mathbf{u}_N]$ with $\mathbf{u}_i \in \mathcal{K}$ such that

$$\mathbf{U}_o = \arg\max_{\mathbf{U}} \operatorname{tr}\left(\mathbf{U}^*\mathbf{S}_\mathcal{K}\mathbf{U}\right) \quad \text{s.t. } \mathbf{U}^*\mathbf{U} = \mathbf{J}. \quad (14)$$

We write the set of projections as a linear combination of samples as $\mathbf{U} = \tilde{\mathbf{X}}_\psi\mathbf{Q}$, and (14) becomes

$$\mathbf{Q}_o = \arg\max_{\mathbf{Q}} \operatorname{tr}\left(\mathbf{Q}^T\tilde{\mathbf{X}}_\psi^T\mathbf{J}\tilde{\mathbf{X}}_\psi\tilde{\mathbf{X}}_\psi^T\mathbf{J}\tilde{\mathbf{X}}_\psi\mathbf{Q}\right) = \arg\max_{\mathbf{Q}} \operatorname{tr}\left(\mathbf{Q}^T\tilde{\mathbf{K}}\tilde{\mathbf{K}}\mathbf{Q}\right) \quad \text{s.t. } \mathbf{Q}^T\tilde{\mathbf{X}}_\psi^T\mathbf{J}\tilde{\mathbf{X}}_\psi\mathbf{Q} = \mathbf{Q}^T\tilde{\mathbf{K}}\mathbf{Q} = \mathbf{J} \quad (15)$$

where $\tilde{\mathbf{K}} = \tilde{\mathbf{X}}_\psi^*\tilde{\mathbf{X}}_\psi$ is the centralized kernel matrix. The eigendecomposition of $\tilde{\mathbf{K}}$ then yields the solution of the above as

$$\tilde{\mathbf{K}} = \mathbf{V}\Lambda\mathbf{V}^T = \mathbf{V}|\Lambda|^{\frac{1}{2}}\mathbf{J}|\Lambda|^{\frac{1}{2}}\mathbf{V}^T \quad (16)$$

where Λ is a diagonal matrix whose main diagonal consists of p positive and q negative eigenvalues ($p + q \leq N$) in the following order: first, positive eigenvalues with decreasing values, then negative ones with decreasing absolute values, and, finally, zero values. Matrix $|\Lambda|$ is the diagonal matrix containing the absolute values of the eigenvalues. The fundamental symmetry, matrix $\mathbf{J}$, is defined as in (4), and (p, q) is the pseudo-Euclidian space's signature. Consequently, we obtain the optimal solution of (15) from $\mathbf{Q}_o = \mathbf{V}_{p+q}|\Lambda_{p+q}|^{-(1/2)}$ and the optimal projection matrix from $\mathbf{U}_o = \tilde{\mathbf{X}}_\psi\mathbf{V}_{p+q}|\Lambda_{p+q}|^{-(1/2)}$, where Λ_{p+q} contains the nonzero eigenvalues and $\mathbf{V}_{p+q}$ denotes the corresponding eigenvectors.

Let $\mathbf{y} \in \mathbb{C}^d$ be a new sample, and $\acute{\mathbf{y}} = \psi(\mathbf{y}) \in \mathcal{K}$ denotes its mapping. Then, the part of $\acute{\mathbf{y}}$ which belongs to the positive subspace $\mathbb{R}^p$ is given by

$$\begin{aligned}\acute{\mathbf{y}}_+ &= |\Lambda_p|^{-\frac{1}{2}}\mathbf{V}_p^T\mathbf{M}^T\mathbf{X}_\psi^*\psi(\mathbf{y}) \\ &= |\Lambda_p|^{-\frac{1}{2}}\mathbf{V}_p^T\mathbf{M}^T\begin{bmatrix}\langle\psi(\mathbf{x}_1), \psi(\mathbf{y})\rangle_\mathcal{K} \\ \cdots \\ \langle\psi(\mathbf{x}_N), \psi(\mathbf{y})\rangle_\mathcal{K}\end{bmatrix} \\ &= |\Lambda_p|^{-\frac{1}{2}}\mathbf{V}_p^T\mathbf{M}^T\begin{bmatrix}k(\mathbf{x}_1, \mathbf{y}) \\ \cdots \\ k(\mathbf{x}_N, \mathbf{y})\end{bmatrix}\end{aligned} \quad (17)$$

where $\Lambda_\mathbf{p}$ contains only the positive eigenvalues, and $\mathbf{V}_p$ denotes the corresponding eigenvectors. Similarly, we can compute the features $\acute{\mathbf{y}}_- \in \mathbb{R}^q$ using

$$\acute{\mathbf{y}}_- = |\Lambda_q|^{-\frac{1}{2}}\mathbf{V}_q^T\mathbf{M}^T\mathbf{X}_\psi^*\psi(\mathbf{y}) \quad (18)$$

where Λ_q and $\mathbf{V}_q$ correspond to the negative eigenvalues. Furthermore, we can verify that the inner product of $\acute{\mathbf{x}}, \acute{\mathbf{y}} \in \mathcal{K}$ is equal to the kernel value as follows:

$$\begin{aligned}\langle\acute{\mathbf{x}}, \acute{\mathbf{y}}\rangle_\mathcal{K} &= \acute{\mathbf{x}}^*\acute{\mathbf{y}} = \acute{\mathbf{x}}^T\mathbf{J}\acute{\mathbf{y}} \\ &= \psi(\mathbf{x})^*\tilde{\mathbf{X}}_\psi\mathbf{V}|\Lambda|^{-\frac{1}{2}}\mathbf{J}|\Lambda|^{-\frac{1}{2}}\mathbf{V}^T\tilde{\mathbf{X}}_\psi^*\psi(\mathbf{y}) \\ &= \psi(\mathbf{x})^T\mathbf{J}\mathbf{U}^*\mathbf{U}\mathbf{J}\psi(\mathbf{y}) = \psi(\mathbf{x})^T\mathbf{J}\psi(\mathbf{y}) \\ &= \langle\psi(\mathbf{x}), \psi(\mathbf{y})\rangle_\mathcal{K} = k(\mathbf{x}, \mathbf{y}).\end{aligned} \quad (19)$$

In order to establish a dimensionality reduction strategy, we can start by expanding the objective function of the optimization problem (14) as

$$\begin{aligned}\operatorname{tr}\left(\mathbf{U}^*\mathbf{S}_\mathcal{K}\mathbf{U}\right) &= \operatorname{tr}\left(\mathbf{Q}^T\tilde{\mathbf{K}}\tilde{\mathbf{K}}\mathbf{Q}\right) \\ &= \operatorname{tr}\left(|\Lambda|^{-\frac{1}{2}}\mathbf{V}^T\mathbf{V}\Lambda\mathbf{V}^T\mathbf{V}\Lambda\mathbf{V}^T\mathbf{V}|\Lambda|^{-\frac{1}{2}}\right) \\ &= \operatorname{tr}(|\Lambda|) = \sum_{i=1}^{N}|\lambda_i|.\end{aligned} \quad (20)$$

As can be observed, the actual functional to be minimized is based on the absolute eigenvalues $|\lambda_i|$. Hence, the dimensionality reduction may be performed by removing the eigenvectors that correspond to the smallest in terms of magnitude eigenvalues. The signature of the reduced Krein space is then given by (p_1, q_1), with $p_1 \leq p$ and $q_1 \leq q$.

B. Direct KPCA in Krein Spaces

In this section, we capitalize on the properties of our kernel in order to define a special version of KPCA in Krein spaces. As we will see in the next section, the proposed KPCA does not require the computation of preimages, and as such we call it direct KPCA. We will then use our direct KPCA as a basis for an exact incremental KPCA in Krein space.

Let $\mathbf{X}_\psi = [\psi(\mathbf{x}_1) \ \cdots \ \psi(\mathbf{x}_N)]$ be the matrix of N known samples in $\mathcal{K}$ (for simplicity we assume zero mean). We define matrix $\mathbf{X}_a = [a(\mathbf{x}_1) \ \cdots \ a(\mathbf{x}_N)]$ and matrix $\mathbf{X}_b = [b(\mathbf{x}_1) \ \cdots \ b(\mathbf{x}_N).]$.[1] From the eigendecomposition of $\mathbf{K} = \mathbf{X}_\psi^* \mathbf{X}_\psi$ we get

$$\mathbf{X}_\psi^* \mathbf{X}_\psi = \mathbf{X}_a^T \mathbf{X}_b = \mathbf{V}_\psi \Lambda_\psi \mathbf{V}_\psi^T. \tag{21}$$

The eigenspace of our KPCA is given by $\mathbf{U}_\psi = \mathbf{X}_\psi \mathbf{V}_\psi |\Lambda_\psi|^{-(1/2)}$ and $\Sigma_\psi = |\Lambda_\psi|^{(1/2)}$ by (16). Let us define $\mathbf{U}_a \triangleq \mathbf{X}_a \mathbf{V}_\psi |\Lambda_\psi|^{-(1/2)}$ and $\mathbf{U}_b \triangleq \mathbf{X}_b \mathbf{V}_\psi |\Lambda_\psi|^{-(1/2)}$. We have $\mathbf{X}_a = \mathbf{U}_a \Sigma_\psi \mathbf{V}_\psi^T$ and $\mathbf{X}_b = \mathbf{U}_b \Sigma_\psi \mathbf{V}_\psi^T$. Additionally, the following properties hold:

$$\begin{aligned}
\mathbf{U}_\psi^* \psi(\mathbf{x}) &= |\Lambda_\psi|^{-\frac{1}{2}} \mathbf{V}_\psi^T \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}) \\ \cdots \\ k(\mathbf{x}_N, \mathbf{x}) \end{bmatrix} \\
&= |\Lambda_\psi|^{-\frac{1}{2}} \mathbf{V}_\psi^T \begin{bmatrix} a(\mathbf{x}_1)^T b(\mathbf{x}) \\ \cdots \\ a(\mathbf{x}_N)^T b(\mathbf{x}) \end{bmatrix} \\
&= |\Lambda_\psi|^{-\frac{1}{2}} \mathbf{V}_\psi^T \mathbf{X}_a^T b(\mathbf{x}) = \mathbf{U}_a^T b(\mathbf{x}) \qquad (22) \\
\mathbf{U}_a^T b(\mathbf{x}) &= |\Lambda_\psi|^{-\frac{1}{2}} \mathbf{V}_\psi^T \begin{bmatrix} a(\mathbf{x}_1)^T b(\mathbf{x}) \\ \cdots \\ a(\mathbf{x}_N)^T b(\mathbf{x}) \end{bmatrix} \\
&= |\Lambda_\psi|^{-\frac{1}{2}} \mathbf{V}_\psi^T \begin{bmatrix} b(\mathbf{x}_1)^T a(\mathbf{x}) \\ \cdots \\ b(\mathbf{x}_N)^T a(\mathbf{x}) \end{bmatrix} \\
&= \mathbf{U}_b^T a(\mathbf{x}) \qquad (23) \\
\mathbf{U}_a^T \mathbf{U}_b &= |\Lambda_\psi|^{-\frac{1}{2}} \mathbf{V}_\psi^T \mathbf{X}_a^T \mathbf{X}_b \mathbf{V}_\psi |\Lambda_\psi|^{-\frac{1}{2}} \\
&= |\Lambda_\psi|^{-\frac{1}{2}} \mathbf{V}_\psi^T \mathbf{X}_\psi^* \mathbf{X}_\psi \mathbf{V}_\psi |\Lambda_\psi|^{-\frac{1}{2}} \\
&= \mathbf{U}_\psi^* \mathbf{U}_\psi = \mathbf{J}. \qquad (24)
\end{aligned}$$

The procedures for computing the eigenspace $\mathbf{U}_a$, $\mathbf{U}_b$, and Σ_ψ are summarized in Algorithm 1.

C. Direct Incremental KPCA

We now show that the proposed direct KPCA in Krein space does not require the computation of preimages (as opposed to the KPCA proposed in [21]). We then capitalize on this property and propose an exact incremental KPCA. More specifically, we show that the computation of preimages is not required for general kernels that satisfy (10).

Let $\mathbf{Y} = [\mathbf{x}_{N+1} \ \cdots \ \mathbf{x}_{N+M}] \in \mathbb{C}^{d \times M}$ be a set of M new observations for the incremental update of our KPCA. $\mathbf{Y}_\psi = [\psi(\mathbf{x}_{N+1}) \ \cdots \ \psi(\mathbf{x}_{N+M})]$ is the data matrix in $\mathcal{K}$. Let us also define $\mathbf{Y}_a = [a(\mathbf{x}_{N+1}) \ \cdots \ a(\mathbf{x}_{N+M})]$ and $\mathbf{Y}_b = [b(\mathbf{x}_{N+1}) \ \cdots \ b(\mathbf{x}_{N+M})]$. Finally, we denote the combined sample matrix by $[\mathbf{X}_\psi \ \ \mathbf{Y}_\psi]$, where $\mathbf{X}_\psi$ is the currently available data in $\mathcal{K}$. The combined matrix is equivalent to

$$\left[\mathbf{U}_\psi \Sigma_\psi \mathbf{V}_\psi^T \ \ \mathbf{U}_\psi \mathbf{U}_\psi^* \mathbf{Y}_\psi + \mathbf{Q}_\psi \mathbf{R}_\psi \right] \tag{25}$$

where $\mathbf{H}_\psi = \mathbf{Y}_\psi - \mathbf{U}_\psi \mathbf{U}_\psi^* \mathbf{Y}_\psi$ is the complement to the $\mathbf{U}_\psi$ subspace, $\mathbf{Q}_\psi$ is an orthogonal matrix, and $\mathbf{Q}_\psi \mathbf{R}_\psi = \mathbf{H}_\psi$ is satisfied. We obtain $\mathbf{Q}_\psi = \mathbf{H}_\psi \Omega |\Delta|^{-(1/2)}$ and $\mathbf{R}_\psi = |\Delta|^{(1/2)} \Omega^T$ by the eigendecomposition of $\mathbf{H}_\psi^* \mathbf{H}_\psi = \Omega \Delta \Omega^T$. We define $\mathbf{H}_a \triangleq \mathbf{Y}_a - \mathbf{U}_a \mathbf{U}_b^T \mathbf{Y}_a$ and $\mathbf{H}_b \triangleq \mathbf{Y}_b - \mathbf{U}_b \mathbf{U}_a^T \mathbf{Y}_b$, and compute the eigendecomposition of $\mathbf{H}_a^T \mathbf{H}_b$ to avoid the computation of the unknown projection of $\mathbf{Y}_\psi$ onto $\mathbf{U}_\psi$

$$\begin{aligned}
\mathbf{H}_a^T \mathbf{H}_b &= (\mathbf{Y}_a^T - \mathbf{Y}_a^T \mathbf{U}_b \mathbf{U}_a^T)(\mathbf{Y}_b - \mathbf{U}_b \mathbf{U}_a^T \mathbf{Y}_b) \\
&= (\mathbf{Y}_\psi - \mathbf{U}_\psi \mathbf{U}_\psi^* \mathbf{Y}_\psi)^* (\mathbf{Y}_\psi - \mathbf{U}_\psi \mathbf{U}_\psi^* \mathbf{Y}_\psi) \\
&= \mathbf{H}_\psi^* \mathbf{H}_\psi = \Omega \Delta \Omega^T. \qquad (26)
\end{aligned}$$

The matrix in (25) can be rewritten as

$$\left[\mathbf{U}_\psi \ \ \mathbf{Q}_\psi \right] \mathbf{L}_\psi \begin{bmatrix} \mathbf{V}_\psi^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{27}$$

where $\mathbf{L}_\psi = \begin{bmatrix} \Sigma_\psi & \mathbf{U}_\psi^* \mathbf{Y}_\psi \\ \mathbf{0} & \mathbf{R}_\psi \end{bmatrix}$. The SVD of $[\mathbf{X}_\psi \ \ \mathbf{Y}_\psi]$ is then given by

$$\left[\left[\mathbf{U}_\psi \ \ \mathbf{Q}_\psi \right] \tilde{\mathbf{U}}_\psi \right] \left[\tilde{\Sigma}_\psi \right] \left[\tilde{\mathbf{V}}_\psi^T \begin{bmatrix} \mathbf{V}_\psi^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \right] \tag{28}$$

where $\mathbf{L}_\psi \overset{\text{svd}}{=} \tilde{\mathbf{U}}_\psi \tilde{\Sigma}_\psi \tilde{\mathbf{V}}_\psi^T$ (which matrix for indefinite kernels may contain positive and negative eigenvalues). Thus, we only need to compute the SVD of $\mathbf{L}_\psi$ for the incremental update of our eigenspace, $\mathbf{U}'_\psi = [\mathbf{U}_\psi \ \ \mathbf{L}_\psi] \tilde{\mathbf{U}}_\psi$ and $\Sigma'_\psi = |\tilde{\Sigma}_\psi|$. As $\mathbf{U}_\psi$ and $\mathbf{H}_\psi$ are not directly given by our KPCA, we define $\mathbf{Q}_a \triangleq \mathbf{H}_a \Omega |\Delta|^{-(1/2)}$ and $\mathbf{Q}_b \triangleq \mathbf{H}_b \Omega |\Delta|^{-(1/2)}$, and set $\mathbf{U}'_a = [\mathbf{U}_a \ \mathbf{Q}_a] \tilde{\mathbf{U}}_\psi$, and $\mathbf{U}'_b = [\mathbf{U}_b \ \mathbf{Q}_b] \tilde{\mathbf{U}}_\psi$. Note that this choice satisfies (22)–(24). Algorithm 2 summarizes the proposed incremental update. Because of our direct approach to KPCA, the storage requirements for the incremental update is of fixed complexity [e.g., $\mathcal{O}(2 * 4d(p + q + M))$ for our kernel], and the complexity of the update is also fixed (e.g., in $\mathcal{O}(4dM^2)$ for our kernel), similar to [6].

Summarizing, we have presented a general approach for incremental KPCA with nonpositive definite kernels in Krein space that can be described by two explicitly given mappings

Algorithm 1 Direct KPCA in Krein Space

Input: The set $\mathbf{X} = \left[\mathbf{x}_1 \ \cdots \ \mathbf{x}_N \right] \in \mathbb{C}^{d \times N}$ of N samples, and two explicit mappings, $a : \mathbb{C}^d \to \mathbb{R}^{4d}$ and $b : \mathbb{C}^d \to \mathbb{R}^{4d}$, that satisfy (10).

Output: The eigenspace $\mathbf{U}_a$, $\mathbf{U}_b$ and Σ_ψ.

1: Compute $\mathbf{X}_a = \left[a(\mathbf{x}_1) \ \cdots \ a(\mathbf{x}_N) \right]$ and $\mathbf{X}_b = \left[b(\mathbf{x}_1) \ \cdots \ b(\mathbf{x}_N) \right]$.
2: Find $\mathbf{V}_\psi$ and Λ_ψ *via* $\mathbf{X}_a^T \mathbf{X}_b = \mathbf{X}_\psi^* \mathbf{X}_\psi = \mathbf{V}_\psi \Lambda_\psi \mathbf{V}_\psi^T$.
3: Set $\mathbf{U}_a = \mathbf{X}_a \mathbf{V}_\psi |\Lambda_\psi|^{-(1/2)}$, $\mathbf{U}_b = \mathbf{X}_b \mathbf{V}_\psi |\Lambda_\psi|^{-(1/2)}$, $\Sigma_\psi = |\Lambda_\psi|^{(1/2)}$.
4: Obtain p_1+q_1-reduced set of $\mathbf{U}_a$ and $\mathbf{U}_b$ by keeping p_1+q_1 largest eigenvalue magnitudes in Σ_ψ.

[1] We should note here that, even though mappings $a(.)$ and $b(.)$ are known, mapping $\psi(.)$ is not known and neither can be explicitly defined, and $a(.) \neq b(.)$.

Algorithm 2 Incremental Update of DIKPCA

Input: The previous eigenspace $\mathbf{U}_a$, $\mathbf{U}_b$ and Σ_ψ, and number of previous samples N, the set of M new samples $\mathbf{Y} = \left[\mathbf{x}_{N+1} \cdots \mathbf{x}_{N+M}\right] \in \mathbb{C}^{d\times M}$ and the two mappings a: $\mathbb{C}^d \to \mathbb{R}^{4d}$ and $b : \mathbb{C}^d \to \mathbb{R}^{4d}$, that satisfy (10).

Output: The updated eigenspace $\mathbf{U}'_a$, $\mathbf{U}'_b$, and Σ'_ψ.

1: Calculate the mappings, $\mathbf{Y}_a$ and $\mathbf{Y}_b$, of $\mathbf{Y}$.
2: Find $\mathbf{H}_a = \mathbf{Y}_a - \mathbf{U}_a\mathbf{U}_b^T\mathbf{Y}_a$ and $\mathbf{H}_b = \mathbf{Y}_b - \mathbf{U}_b\mathbf{U}_a^T\mathbf{Y}_b$.
3: Compute $\mathbf{H}_a^T\mathbf{H}_b = \mathbf{H}_\psi^*\mathbf{H}_\psi = \Omega\Delta\Omega^T$ and set $\mathbf{R}_\psi = |\Delta|^{\frac{1}{2}}\Omega^T$, $\mathbf{Q}_a = \mathbf{H}_a\Omega|\Delta|^{-\frac{1}{2}}$ and $\mathbf{Q}_b = \mathbf{H}_b\Omega|\Delta|^{-\frac{1}{2}}$.
4: Form $\mathbf{L}_\psi = \begin{bmatrix} |\Sigma_\psi| & \mathbf{U}_b^T\mathbf{Y}_a \\ \mathbf{0} & \mathbf{R}_\psi \end{bmatrix}$ and compute $\mathbf{L}_\psi \overset{\text{svd}}{=} \tilde{\mathbf{U}}_\psi\tilde{\Sigma}_\psi\tilde{\mathbf{V}}_\psi^T$.
5: Set $\mathbf{U}'_a = [\,\mathbf{U}_a\ \mathbf{Q}_a\,]\,\tilde{\mathbf{U}}_\psi$, $\mathbf{U}'_b = \left[\mathbf{U}_b\ \mathbf{Q}_b\right]\tilde{\mathbf{U}}_\psi$ and $\Sigma'_\psi = \tilde{\Sigma}_\psi$.
6: Obtain p_1+q_1-reduced set of $\mathbf{U}_a$ and $\mathbf{U}_b$ *via* p_1+q_1 largest eigenvalue magnitudes in $|\Sigma'_\psi|$.

Algorithm 3 DIKT at Time t

Input: The previous eigenspace $\mathbf{U}_{a_{t-1}}$, $\mathbf{U}_{b_{t-1}}$, $\Sigma_{\psi_{t-1}}$, locations $\mathbf{A}_{t-1}^{1:P}$, weights $w_{t-1}^{1:P}$, image frame $\mathbf{I}_t \in [0,1]$ and the explicit mappings $a(.)$ and $b(.)$ of the kernel.

1: Draw P particles $\mathbf{A}_t^{1:P}$ from $p(A_t^i|A_{t-1}^{1:P})$ as in (29).
2: Take all image patches from $\mathbf{I}_t$ which correspond to particles $\mathbf{A}_t^{1:P}$, compute their gradients $\mathbf{G}^{1:P}$ and order them lexicographically to form vectors $\mathbf{y}_t^{1:P}$.
3: Compute the probability of each particle $p(\psi(\mathbf{y}_t^i)|\mathbf{A}_t^i)$ as (30) and extract the weights $w_t^{1:P}$.
4: Choose $\mathbf{A}_t^{\text{best}}$ and $\mathbf{y}_t^{\text{best}}$ as the affine transform and features of the particle with the largest weight.
5: Using $\mathbf{y}_t^{\text{best}}$ update subspace by applying Algorithm 2 in a batch after a certain number of frames (5 in our implementation).

$a(.)$ and $b(.)$ such that (10) holds. We coin our approach direct incremental KPCA (DIKPCA). DIKPCA allows the fast update of the eigenspace. In contrast to the incremental version of KPCA proposed in [21], which deals with positive definite kernels, our approach uses a class of special indefinite kernels, which renders finding preimages unnecessary. Therefore, our method is not only faster but also exact.

IV. Visual Tracking

We use our robust and efficient online learning framework for visual tracking. In particular, we combine our appearance model with a motion affine transformation in a particle filter framework, in a similar fashion to [6], [21], and [38].

Generally, a particle filter calculates the posterior of a system's states based on a transition model and an observation model. In our tracking framework, the transition model is described as a Gaussian mixture model around an approximation of the state posterior distribution of the previous time step

$$p\left(A_t^i|A_{t-1}^{1:P}\right) = \sum_{i=1}^{P} w_{t-1}^i \mathcal{N}\left(A_t; A_{t-1}^i, \Xi\right) \tag{29}$$

where A_t^i is the affine transformation of particle i at time t, $A_{t-1}^{1:P}$ is the set of P transformations of the previous time step whose weights are denoted by $w_{t-1}^{1:P}$, and Ξ is an independent covariance matrix. In the first phase, P particles are drawn from (29). In the second phase, the observation model is applied to estimate the weighting for the next iteration (the weights are normalized to ensure $\sum_{i=1}^{P} w_t^i = 1$). Furthermore, the most probable sample is selected as the state A_t^{best} at time t. Thus, the estimation of the posterior distribution is an incremental process and utilizes a hidden Markov model which only relies on the previous time step.

Our observation model computes the probability of a sample being generated by the learned eigenspace in the appearance model. We also assume the probability of the observation in Krein space, given the tracking parameters at t, to be analogous to an exponential as

$$\begin{aligned} p(\psi(\mathbf{y}_t^i)|\mathbf{A}_t^i) &\propto e^{-\gamma\,|(\psi(\mathbf{y}_t^i)-\mathbf{U}_\psi\mathbf{U}_\psi^*\psi(\mathbf{y}_t^i))^*(\psi(\mathbf{y}_t^i)-\mathbf{U}_\psi\mathbf{U}_\psi^*\psi(\mathbf{y}_t^i))|} \\ &= e^{-\gamma\,|(a(\mathbf{y}_t^i)^T-a(\mathbf{y}_t^i)^T\mathbf{U}_b\mathbf{U}_a^T)(b(\mathbf{y}_t^i)-\mathbf{U}_b\mathbf{U}_a^T b(\mathbf{y}_t^i))|} \end{aligned} \tag{30}$$

where $\mathbf{y}_t^i$ is the observation vector at time t of location $\mathbf{A}_t^i$ and γ is the parameter that controls the spread. Note: the distribution can be calculated via $a(.)$, $b(.)$, $\mathbf{U}_a$, and $\mathbf{U}_b$ to avoid the unknown subspace $\mathbf{U}_\psi$. Algorithm 3 describes the proposed visual tracking framework, which we coin direct incremental KPCA tracker (DIKT).

V. Results and Discussion

We pursue our evaluation in two stages. First, we test the performance of DIKT against other state-of-the-art holistic online tracking algorithms. Second, we evaluate the general robustness of our KPCA framework DKPCA with our kernel, and compare it to standard PCA and KPCA in Hilber space with a Gaussian RBF kernel.

A. Object Tracking

In this section, we present performance evaluation results of the proposed DIKT. We also compare the performance of our method with that of four other state-of-the-art tracking approaches.

1) IVT [6], the MATLAB implementation of which is publicly available at http://www.cs.toronto.edu/~dross/ivt/.
2) IKPCA [21], the MATLAB implementation of the IKPCA was kindly provided by the authors of this paper.
3) L1 tracker proposed in [34], the implementation of which is publicly available at http://www.ist.temple.edu/~hbling/code_data.htm.
4) MIL tracker [45], the implementation of which (only for translation motion model) is publicly available at http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml and which we carefully modified it in order to support an affine motion model in a particle filter framework.

We evaluate the performance of all methods on nine very popular video sequences, V_i, $i = 1, \ldots, 9$ (subsets of which are used in [6], [34], and [45]–[47]). The videos contain

drastic changes of the target's appearance, including pose variation, occlusions, and nonuniform illumination. Representative frames of the video sequences are illustrated in Fig. 1, while the videos of the tracking results can be found at http://www.doc.ic.ac.uk/~sl609/dikt/. We use videos $V_1 - V_5$ to illustrate performance for the application of face tracking. Tracking of vehicles is assessed using V_6 and V_7. Finally, other objects are tracked in V_8 and V_9. Videos V_4 and V_5 are available at http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml and the remaining videos are published at http://www.cs.toronto.edu/~dross/ivt/.

Video V_1 is provided along with seven annotated points which indicate the ground truth. We also annotate 3–7 fiducial points for the remaining sequences. As usual, our quantitative performance evaluation is based on the root mean square (RMS) errors between the true and the estimated locations of these points [6]. Similar to [45], we additionally present precision plots that visualize the quality of the tracking. Such graphs show the percentage of frames in which the target was tracked with an RMS error less than a certain threshold.

In our experiments, all trackers use an affine motion model with a fixed number of drawn particles (800 particles). In the following, we present the results obtained for two versions of our experimental settings. In the first version, we attempt to optimize the performance of all trackers using video-specific parameters. That is, for each tracker and video, we found the parameters that gave the best performance in terms of robustness (i.e., how many times the tracker went completely off) and accuracy (measured by the RMS error). In the second and most interesting version, we present results by keeping the parameters of each tracker fixed for all videos. Again, we use the parameters that gave the best performance in terms of robustness and accuracy.

1) Tracking With Video-Specific Parameters: We denote by DIKT-specific, IVT-specific, IKPCA-specific, L1-specific, and MIL-specific the video-specific versions of the trackers. The optimization criterion was the minimization of the RMS error between the true and the estimated location of the points. Apart from the L1-specific tracker (for which the resolution of the template increases geometrically the complexity), the tracking template was chosen to be of resolution of 32×32. For all trackers, we optimized with respect to (wrt) the variance of the Gaussian from which we sample the particles. Expect for the variance of the Gaussian, which is common for all the systems, we optimize.

1) For DIKT-specific, IVT-specific, and IKPCA-specific wrt the number of components (which ranged between 16 and 18) and the variance of the exponential that models the probability of a sample being generated by the learned subspace. For IKPCA-specific we also optimized wrt the radius of the GRBF function.
2) For L1-specific wrt the number of templates and the resolution of the template (actually, the tracking becomes impractical when choosing templates of resolution more than 20×20).
3) For MIL-specific wrt the parameters mentioned in [45] (i.e., the number of positives in each frame, the number that controls the sampling of negative examples, the learning rate for the weak classifiers, etc).

TABLE I
MEAN RMS ERROR FOR VIDEO-SPECIFIC TRACKING. "(LOST)" INDICATES SEQUENCES IN WHICH THE TRACKER CLEARLY DOES NOT FOLLOW THE TARGET THROUGHOUT

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9
IVT-specific	6.82	(Lost)	4.07	10.79	(Lost)	**3.31**	1.78	2.62	(Lost)
IKPCA-specific	(Lost)	(Lost)	(Lost)	(Lost)	(Lost)	(Lost)	(Lost)	(Lost)	(Lost)
L1-specific	6.17	(Lost)	2.87	11.10	12.68	9.53	**1.62**	13.58	(Lost)
MIL-specific	16.95	(Lost)	13.61	14.62	37.56	12.73	4.14	23.87	17.62
DIKT-specific	**4.48**	**2.27**	**2.49**	**5.62**	**11.28**	3.40	1.80	**1.96**	**5.90**

TABLE II
MEAN RMS ERROR FOR GENERAL TRACKING. "(LOST)" INDICATES SEQUENCES IN WHICH THE TRACKER CLEARLY DOES NOT FOLLOW THE TARGET THROUGHOUT

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9
IVT	8.13	(Lost)	4.13	**13.14**	(Lost)	27.79	2.02	24.48	(Lost)
IKPCA	(Lost)	(Lost)	(Lost)	(Lost)	(Lost)	(Lost)	(Lost)	(Lost)	(Lost)
L1	(Lost)	(Lost)	2.87	(Lost)	**12.94**	(Lost)	**1.67**	39.15	(Lost)
MIL	51.36	(Lost)	13.61	17.78	38.19	(Lost)	4.14	40.80	(Lost)
DIKT	**4.81**	**2.43**	**2.49**	(Lost)	(Lost)	**3.51**	2.15	**2.26**	(Lost)

The tracking rates for the tested systems on a desktop running an Intel Core i7 870 at 2.93 GHz with 8 GB of RAM and MATLAB 64 are as follows: for IVT 4 frames/s, for DIKT 3 frames/s, for IKPCA 0.7–1 frame/s, for MIL 0.25 frames/s, and for L1 less than 0.1 frames/s.

For these versions of the trackers, Table I lists the mean RMS error for all sequences, while Fig. 2 plots the RMS error as a function of the frame number. Fig. 3 shows the accuracy in terms of precision plots. Qualitative tracking results for all methods are shown in Fig. 1. Finally, videos and sample code for DIKT may be found at http://www.doc.ic.ac.uk/~sl609/dikt/.

In general, DIKT-specific outperforms all other trackers in terms of robustness, accuracy, and precision. In terms of robustness, DIKT-specific successfully follows the target for all sequences, including V_2, where all other trackers fail. In terms of RMS error, DIKT-specific achieves by far the best results for all videos with the exception of video V_7, where it is slightly outperformed by IVT-specific and L1-specific. While MIL-specific appears to be robust (it loses the target for one video sequence only), it is generally not as precise as the other trackers. Clearly, IKPCA is inferior to the other trackers. We believe that this performance degradation is induced by the search for preimages, which accumulates errors and eventually makes the tracker go off in prolonged and challenging video sequences.

2) Tracking With Fixed Parameters: We denote by DIKT, IVT, IKPCA, L1, and MIL the versions of the trackers that use fixed parameters for all videos. The parameters were optimized in two videos we recorded, which are not included in the test

Fig. 1. Example frames of the tracking with video-specific parameters for V_1–V_9 (top to bottom). The results of DIKT-specific (◯) versus IVT-specific (□), IKPCA-specific (◇), L1-specific (▷), and MIL-specific (◁) are shown. The ground truth is indicated by ×. The tracked area of DIKT-specific is visualized by a magenta bounding box.

set. The number of components for DIKT and IVT was set to 16, and the resolution for all trackers except for L1 was set to 32×32. For all the test videos, we list the mean RMS errors in Table II, while Figs. 4 and 5 illustrate the accuracy and precision of all trackers. The video results can be found at http://www.doc.ic.ac.uk/~sl609/dikt/.

Compared to the performance of all other trackers, that of DIKT appears to be significantly less affected by video-specific parameter fine-tuning. In terms of robustness, DIKT is still among the most robust trackers, because in six out of nine videos the target was successfully tracked. In terms of accuracy and precision, DIKT appears to be the only tracker that performs almost equally well to its video-specific version (DIKT-specific).

B. Face Recognition

In this section, we evaluate the performance of DKPCA for face recognition with illumination changes and pose variations (as the number of training sample is small, we apply the batch

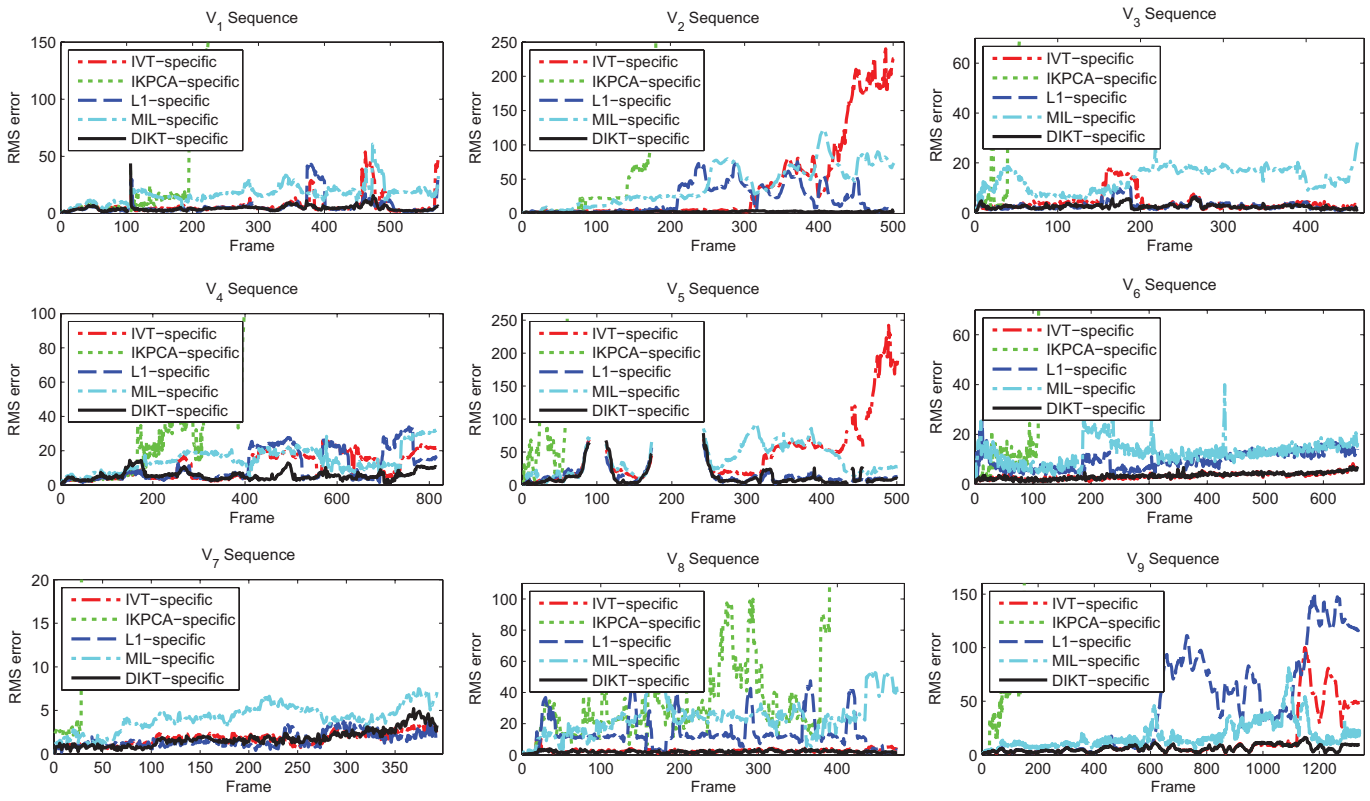


Fig. 2. RMS error for each frame of the tuned DIKT-specific against IVT-specific, IKPCA-specific, L1-specific, and MIL-specific for V_1–V_9 (left to right, top to bottom). There is no RMS error during complete occlusions.

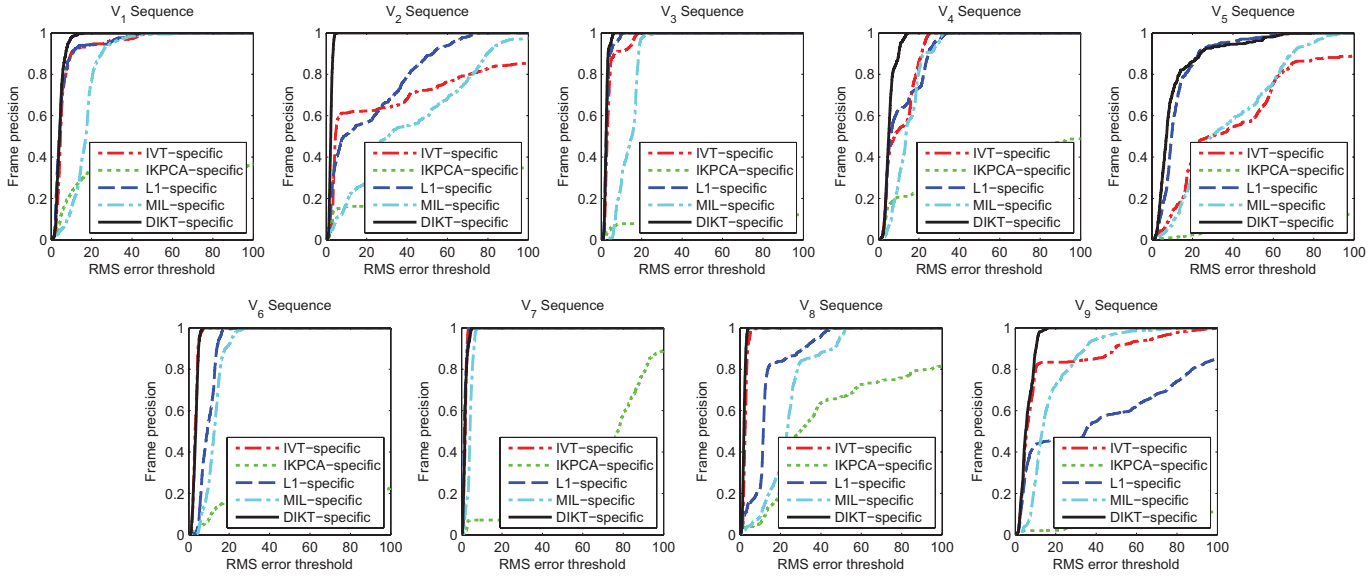


Fig. 3. Frame precision plots (showing the percentage of frames in which the target was tracked with an RMS error less than a certain threshold) for the tuned DIKT-specific against IVT-specific, IKPCA-specific, L1-specific, and MIL-specific for V_1–V_9 (left to right, top to bottom).

version of our method as seen in Algorithm 1). We compare the results to standard PCA and KPCA with a Gaussian RBF kernel. We optimize the Gaussian RBF kernel's deviation for each experiment.

1) Extended Yale B Database: The extended Yale B database [48] contains 16 128 images of 38 subjects under 9 poses and 64 illumination conditions. We use a subset that consists of 64 near-frontal images for each subject. For training, we randomly select a subset with 5, 10, or 20 images per subject. For testing, we utilize the remaining images. Finally, we perform 20 different random realizations of the training/test sets. Table III shows the average recognition rate of the tested methods. DKPCA performs best for all setups.

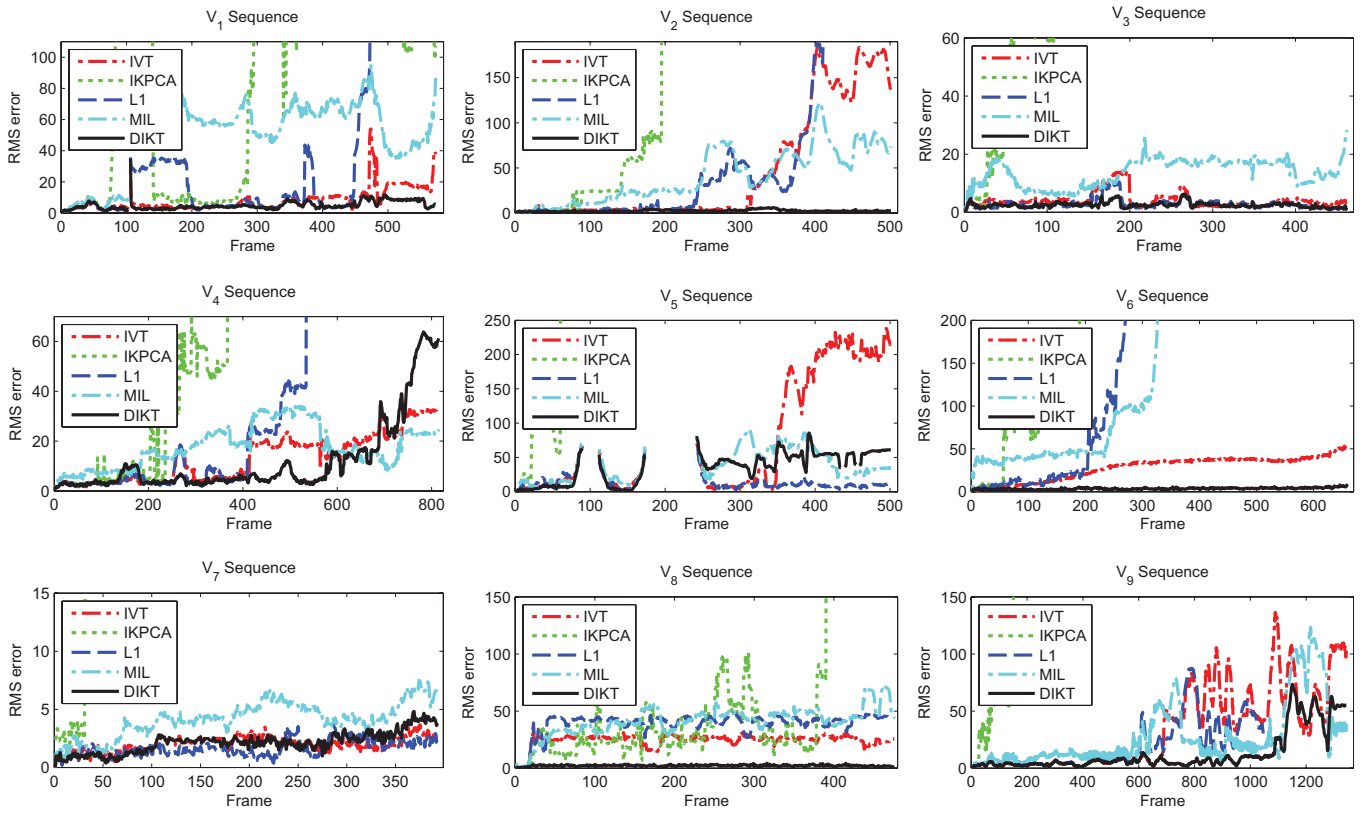


Fig. 4. RMS error for each frame of DIKT against IVT, IKPCA, L1, and MIL for $V_1 - V_9$ (left to right, top to bottom). There is no RMS error during complete occlusions.

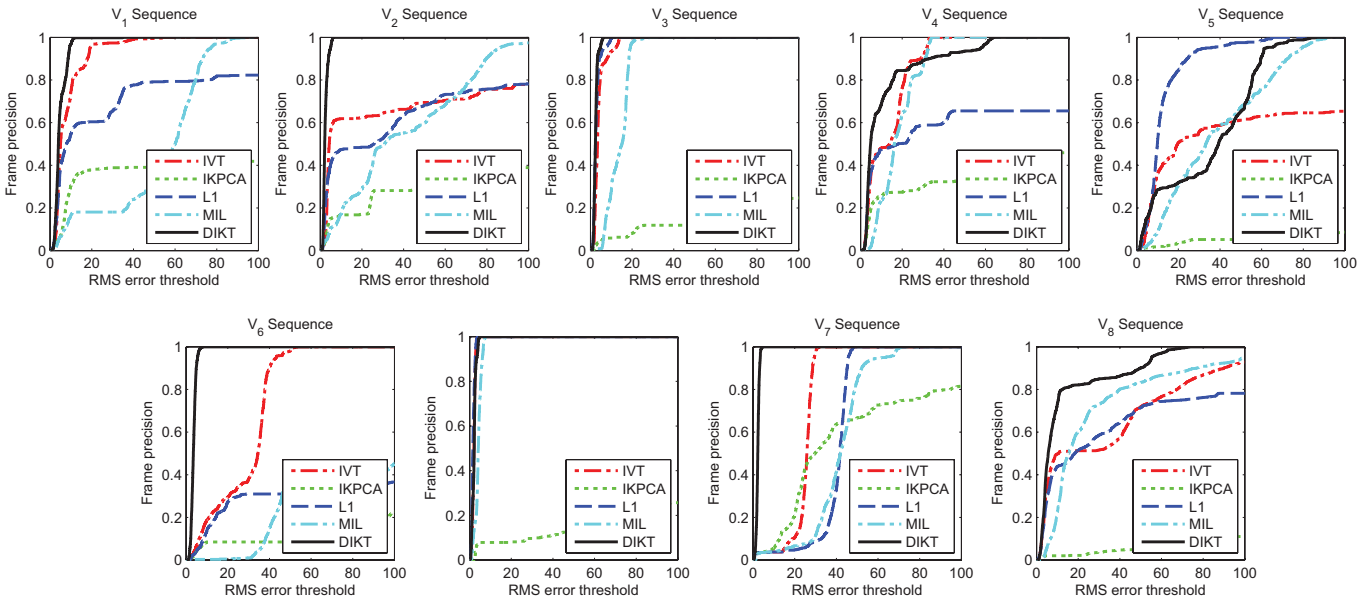


Fig. 5. Frame precision plots (showing the percentage of frames in which the target was tracked with an RMS error less than a certain threshold) for DIKT against IVT, IKPCA, L1, and MIL for V_1–V_9 (left to right, top to bottom).

Furthermore, we train each method with the same random selection of three classes, each class with the same five random training images. After computing the subspace of each algorithm (linear PCA, KPCA with Gaussian RBF kernel, and the proposed DKPCA), we project the samples of the test set onto it (the deviation of the RBF kernel was set to the one that gave the best performance in the recognition experiment). Fig. 6 plots the features corresponding to the two largest eigenvalues along with some ellipsis which describes the distribution of each facial class. Standard PCA and KPCA with Gaussian RBF cluster the training data poorly. Our DKPCA is more successful in separating the classes.

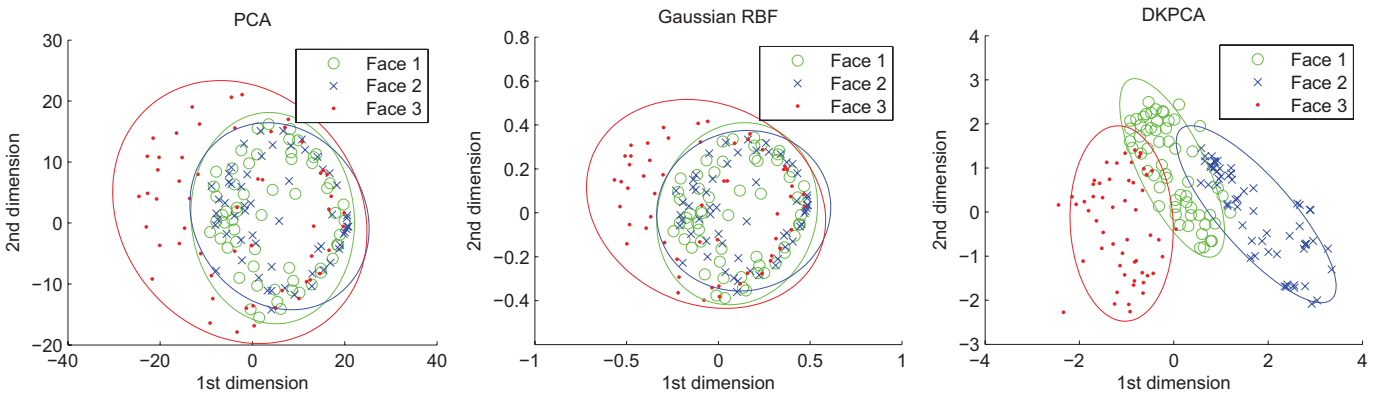


Fig. 6. Test data, as projected by the learned subspaces of PCA, Gaussian RBF, and DKPCA (left to right). We train with three randomly selected classes (i.e., subjects) in the usual manner (trained on five). We then plot the corresponding projections of the test samples.

TABLE III
AVERAGE RECOGNITION RATE WITH YALE DATABASE

	Trained on 5	Trained on 10	Trained on 20
PCA	31.1%	45.2%	59.3%
Gaussian RBF	31.6%	45.5%	59.4%
DKPCA	**62.1%**	**77.7%**	**88.4%**

TABLE IV
AVERAGE RECOGNITION RATE WITH CMU PIE DATABASE

	Trained on 5	Trained on 10
PCA	26.1%	39.1%
Gaussian RBF	26.1%	39.1%
DKPCA	**52.3%**	**69.4%**

The enhanced class-separability achieved by the proposed method can be explained by the metric multidimensional scaling (MMS) perspective of PCA, which can be provided through [49] and [50]. Under this perspective, standard ℓ_2 PCA finds the optimal linear projections that best preserve the ℓ_2 distances. As is well known, these distances can be arbitrarily biased by the presence of outliers (the same holds for KPCA with Gaussian RBF kernels). Therefore, in the presence of outliers, PCA and KPCA with Gaussian RBF are not suitable for providing a consistent way of measuring distances in a facial class. On the other hand, under the MMS perspective, the proposed DKPCA finds the optimal linear projections that best preserve the proposed robust distance. And because, as we argue in our paper, this distance is robust to outliers, the proposed DKPCA provides a more consistent way of representing the samples in a facial class.

2) CMU PIE Database: Our final tests are conducted with the CMU PIE database [51]. The dataset consists of more than 41 000 face images of 68 subjects. The database contains faces under varying pose, illumination, and expression. We use the near frontal poses (C05, C07, C09, C27, and C29) and a total of 170 images for each subject. For training, we randomly selected a subset with 5 or 10 images per subject. For testing, we utilize the remaining images. Finally, we perform 20 different random realizations of the training and test sets. The results are shown in Table IV.

VI. CONCLUSION

We proposed a robust online kernel learning framework for efficient visual tracking. We used a nonlinear appearance model learned via KPCA and a robust gradient-based kernel. As this kernel is not semipositive definite, we showed how to extend the KPCA formulation into Krein spaces. Finally, we showed that our kernel has a very special form which enables us to formulate a direct version of KPCA in Krein space and does not require the calculation of preimages. Based on this property, we then proposed an efficient and exact incremental KPCA. By combining our appearance model with a particle filter, the proposed tracking framework achieved state-of-the-art performance in many popular difficult tracking scenarios. We showed further applications of our kernel framework by testing on face recognition, for which we improve upon ℓ_2-norm PCA and KPCA in Hilbert space with a Gaussian RBF kernel.

In future work, we intend to apply our kernel framework to other applications which would benefit from its robustness. In relation to tracking, we plan to investigate the influence of the forgetting factor to our incremental KPCA in Krein space.

APPENDIX
KERNEL PROPERTIES

Let kernel $k : \mathbb{C}^d \times \mathbb{C}^d \to \mathbb{R}$ be the kernel from (7) and $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{C}^d$ are two samples.

Lemma 1: $\sum_{c=1}^{d} \mathbf{R}(c) \leq \sqrt{d}\sqrt{\sum_{c=1}^{d} \mathbf{R}^2(c)}$.

Proof: We show for any d-dimensional $\mathbf{R} \in \mathbb{R}^+$

$$\sum_{c=1}^{d} \mathbf{R}(c) \leq \sqrt{d}\sqrt{\sum_{c=1}^{d} \mathbf{R}^2(c)}$$

$$\Leftrightarrow \sum_{c=1}^{d} \mathbf{R}(c) \sum_{c=1}^{d} \mathbf{R}(c) \leq d \sum_{c=1}^{d} \mathbf{R}^2(c)$$

$$\Leftrightarrow \sum_{c=1}^{d} \mathbf{R}^2(c) + 2\sum_{c=1}^{d-1} \sum_{e=c+1}^{d} \mathbf{R}(c)\mathbf{R}(e)$$

$$\leq \sum_{c=1}^{d} \mathbf{R}^2(c) + \sum_{c=1}^{d} \mathbf{R}^2(c)(d-1)$$

$$\Leftrightarrow 0 \leq \sum_{c=1}^{d} \mathbf{R}^2(c)(d-1) - 2\sum_{c=1}^{d-1}\sum_{e=c+1}^{d} \mathbf{R}(c)\mathbf{R}(e)$$

$$= \sum_{c=1}^{d-1}\sum_{e=c+1}^{d} (\mathbf{R}(c) - \mathbf{R}(e))^2. \quad (31)$$

■

Remark 1: $|k(\mathbf{x}_i, \mathbf{x}_j)| \leq 1$.

Proof: With Lemma 1, the following holds:

$$|k(\mathbf{x}_i, \mathbf{x}_j)| = \left| \frac{\sum_{c=1}^{d} \mathbf{R}_i(c)\cos(\Delta\theta(c))}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_i^2(c)d}} + \frac{\sum_{c=1}^{d} \mathbf{R}_j(c)\cos(\Delta\theta(c))}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_j^2(c)d}} \right|$$

$$\leq \left| \frac{\sum_{c=1}^{d} \mathbf{R}_i(c)\cos(\Delta\theta(c))}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_i^2(c)d}} \right| + \left| \frac{\sum_{c=1}^{d} \mathbf{R}_j(c)\cos(\Delta\theta(c))}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_j^2(c)d}} \right|$$

$$\leq \left| \frac{\sum_{c=1}^{d} \mathbf{R}_i(c)}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_i^2(c)d}} \right| + \left| \frac{\sum_{c=1}^{d} \mathbf{R}_j(c)}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_j^2(c)d}} \right|$$

$$\leq \frac{\sqrt{d}}{2\sqrt{d}} + \frac{\sqrt{d}}{2\sqrt{d}} = 1. \quad (32)$$

■

Remark 2: $l^2(\mathbf{x}_i, \mathbf{x}_j) \geq 0$.

Proof: The distance is denoted by $l^2 : \mathbb{C}^d \times \mathbb{C}^d \to \mathbb{R}$

$$l^2(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_i) - 2k(\mathbf{x}_i, \mathbf{x}_j) + k(\mathbf{x}_j, \mathbf{x}_j)$$

$$= \frac{2\sum_{c=1}^{d} \mathbf{R}_i(c)}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_i^2(c)d}} - 2\left(\frac{\sum_{c=1}^{d} \mathbf{R}_i(c)\cos(\Delta\theta(c))}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_i^2(c)d}} + \frac{\sum_{c=1}^{d} \mathbf{R}_j(c)\cos(\Delta\theta(c))}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_j^2(c)d}} \right) + \frac{2\sum_{c=1}^{d} \mathbf{R}_j(c)}{2\sqrt{\sum_{c=1}^{d} \mathbf{R}_j^2(c)d}}$$

$$\geq \frac{\sum_{c=1}^{d} \mathbf{R}_i(c)}{\sqrt{\sum_{c=1}^{d} \mathbf{R}_i^2(c)d}} - \frac{\sum_{c=1}^{d} \mathbf{R}_i(c)}{\sqrt{\sum_{c=1}^{d} \mathbf{R}_i^2(c)d}} - \frac{\sum_{c=1}^{d} \mathbf{R}_j(c)}{\sqrt{\sum_{c=1}^{d} \mathbf{R}_j^2(c)d}} + \frac{\sum_{c=1}^{d} \mathbf{R}_j(c)}{\sqrt{\sum_{c=1}^{d} \mathbf{R}_j^2(c)d}} = 0. \quad (33)$$

■

REFERENCES

[1] H. He, S. Chen, K. Li, and X. Xu, "Incremental learning from stream data," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 1901–1914, Dec. 2011.

[2] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, pp. 1517–1531, Oct. 2011.

[3] J. Weng and W. Hwang, "Incremental hierarchical discriminant regression," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 397–415, Mar. 2007.

[4] G. Zhou, Z. Yang, S. Xie, and J. Yang, "Online blind source separation using incremental nonnegative matrix factorization with volume constraint," *IEEE Trans. Neural Netw.*, vol. 22, no. 4, pp. 550–560, Apr. 2011.

[5] A. Levy and M. Lindenbaum, "Squential Karhunen–Loeve basis extraction and its application to images," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1371–1374, Aug. 2000.

[6] D. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vision*, vol. 77, nos. 1–3, pp. 125–141, 2008.

[7] J. Weng, Y. Zhang, and W. Hwang, "Candid covariance-free incremental principal component analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 1034–1040, Aug. 2003.

[8] S. Ozawa, S. Pang, and N. Kasabov, "Incremental learning of chunk data for online pattern classification systems," *IEEE Trans. Neural Netw.*, vol. 19, no. 6, pp. 1061–1074, Jun. 2008.

[9] Y. Freund and R. Schapire, "Large margin classification using the perceptron algorithm," *Mach. Learn.*, vol. 37, no. 3, pp. 277–296, 1999.

[10] K. Crammer and Y. Singer, "Ultraconservative online algorithms for multiclass problems," *J. Mach. Learn. Res.*, vol. 3, pp. 951–991, Mar. 2003.

[11] K. Crammer, J. Kandola, and Y. Singer, "Online classification on a budget," in *Proc. 16th Adv. Neural Inf. Process. Syst.*, 2003, pp. 225–232.

[12] J. Kivinen, A. Smola, and R. Williamson, "Online learning with kernels," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2165–2176, Aug. 2004.

[13] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2275–2285, Aug. 2004.

[14] F. Desobry, M. Davy, and C. Doncarli, "An online kernel change detection algorithm," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2961–2974, Aug. 2005.

[15] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast kernel classifiers with online and active learning," *J. Mach. Learn. Res.*, vol. 6, pp. 1579–1619, Sep. 2005.

[16] K. Kim, M. Franz, and B. Schölkopf, "Iterative kernel principal component analysis for image modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 9, pp. 1351–1366, Sep. 2005.

[17] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *J. Mach. Learn. Res.*, vol. 7, pp. 551–585, Dec. 2006.

[18] S. Smale and Y. Yao, "Online learning algorithms," *Found. Comput. Math.*, vol. 6, no. 2, pp. 145–170, 2006.

[19] Y. Ying and D. Zhou, "Online regularized classification algorithms," *IEEE Trans. Inform. Theory*, vol. 52, no. 11, pp. 4775–4788, Nov. 2006.

[20] P. Laskov, C. Gehl, S. Krüger, and K. Müller, "Incremental support vector learning: Analysis, implementation and applications," *J. Mach. Learn. Res.*, vol. 7, pp. 1909–1936, Dec. 2006.

[21] T. Chin and D. Suter, "Incremental kernel principal component analysis," *IEEE Trans. Image Process.*, vol. 16, no. 6, pp. 1662–1674, Jun. 2007.

[22] S. Günter, N. Schraudolph, and S. Vishwanathan, "Fast iterative kernel principal component analysis," *J. Mach. Learn. Res.*, vol. 8, pp. 1893–1918, Dec. 2007.

[23] K. Slavakis, S. Theodoridis, and I. Yamada, "Online kernel-based classification using adaptive projection algorithms," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 2781–2796, Jul. 2008.

[24] W. Liu, I. Park, Y. Wang, and J. C. Príncipe, "Extended kernel recursive least squares algorithm," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3801–3814, Oct. 2009.

[25] F. Orabona, J. Keshet, and B. Caputo, "Bounded kernel-based online learning," *J. Mach. Learn. Res.*, vol. 10, pp. 2643–2666, Nov. 2009.

[26] C. Richard, J. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1058–1067, Mar. 2009.

[27] Y. Yao, "On complexity issues of online learning algorithms," *IEEE Trans. Inform. Theory*, vol. 56, no. 12, pp. 6470–6481, Dec. 2010.

[28] T. Chin, K. Schindler, and D. Suter, "Incremental kernel SVD for face recognition with image sets," in *Proc. Autom. Face Gesture Recog. 7th Int. Conf.*, 2006, pp. 461–466.

[29] M. Black and A. Jepson, "EigenTracking: Robust matching and tracking of articulated objects using a view-based representation," *Int. J. Comput. Vision*, vol. 26, no. 1, pp. 63–84, 1998.

[30] G. Hager and P. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 10, pp. 1025–1039, Oct. 1998.

[31] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, Jan. 2001.

[32] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 810–815, Jun. 2004.

[33] S. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1491–1506, Nov. 2004.
[34] X. Mei and H. Ling, "Robust visual tracking using l1 minimization," in *Proc. 12th IEEE Int. Conf. Comput. Vision*, 2009, pp. 1436–1443.
[35] J. Buenaposada and L. Baumela, "Real-time tracking and estimation of plane pose," in *Proc. 16th Int. Conf. Pattern Recog.*, 2002, pp. 697–700.
[36] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Vision Pattern Recog. Comput. Soc. Conf.*, 2001, pp. 511–518.
[37] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
[38] S. Nikitidis, S. Zafeiriou, and I. Pitas, "Camera motion estimation using a novel online vector field model in particle filters," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 8, pp. 1028–1039, Aug. 2008.
[39] G. Tzimiropoulos, V. Argyriou, S. Zafeiriou, and T. Stathaki, "Robust FFT-based scale-invariant image registration with image gradients," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 10, pp. 1899–1906, Oct. 2010.
[40] E. Pękalska and B. Haasdonk, "Kernel discriminant analysis for positive definite and indefinite kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1017–1032, Jun. 2009.
[41] E. Pękalska and R. Duin, *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*. Singapore: World Scientific, 2005.
[42] B. Hassibi, A. Sayed, and T. Kailath, "Linear estimation in krein spaces. I. Theory," *IEEE Trans. Autom. Control*, vol. 41, no. 1, pp. 18–33, Jan. 1996.
[43] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Subspace learning from image gradient orientations," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012, to be published.
[44] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Robust and efficient parametric face alignment," in *Proc. 13th IEEE Int. Conf. Comput. Vision*, Nov. 2011, pp. 1847–1854.
[45] B. Babenko, M. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.
[46] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. IEEE Comput. Vision Pattern Recogn. Conf.*, Jun. 2009, pp. 983–990.
[47] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.
[48] K. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 684–698, May 2005.
[49] E. Pękalska, P. Paclik, and R. Duin, "A generalized kernel approach to dissimilarity-based classification," *J. Mach. Learn. Res.*, vol. 2, no. 2, pp. 175–211, 2002.
[50] I. Borg and P. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. New York: Springer-Verlag, 2005.
[51] T. Sim, S. Backer, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1615–1618, Dec. 2003.

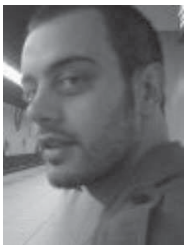
Stephan Liwicki (S'11) received the B.Sc. degree in computer science with a focus on artificial intelligence from the University of Leeds, Leeds, U.K., in 2009, and the M.Phil. degree in computer speech, text, and internet technology from the University of Cambridge, Cambridge, U.K., in 2010, for which he was awarded departmental funding as one of the most promising applicants. He is currently pursuing the Ph.D. degree in computer vision and machine learning with Imperical College London, London, U.K., for which he received one of the competitive Doctoral Training Accounts Studentships from the Engineering and Physical Science Research Council.

He joined the Intelligent Behaviour Understanding Group, Imperial College London, London, U.K., in 2010, as a Research Assistant. His current research interests include human facial tracking and robust online learning with nonlinear subspaces.

Stefanos Zafeiriou (M'09) received the B.Sc. and Ph.D. degrees (Hons.) in informatics from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2003 and 2007, respectively.

He is currently a Lecturer with the Department of Computing, Imperial College London, London, U.K., where he was awarded one of the prestigious Junior Research Fellowships. He received various scholarships and awards during his undergraduate, Ph.D. and post-doctoral studies. He has co-authored more than 60 technical papers, including more than 30 papers in the most prestigious journals, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the *International Journal of Computer Vision*, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, the IEEE TRANSACTIONS ON NEURAL NETWORKS, *Data Mining and Knowledge Discovery*, and *Pattern Recognition*.

Dr. Zafeiriou is an Associate Editor of the *Image and Vision Computing Journal* and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS. He has served as a Program Committee Member for a number of the IEEE international conferences.



Georgios Tzimiropoulos (M'05) received the Diploma degree in electrical and computer engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2004, and the M.Sc. and Ph.D. degrees in signal processing and computer vision from Imperial College London, London, U.K., in 2005 and 2009, respectively.

He is currently a Lecturer (Assistant Professor) with the School of Computer Science, University of Lincoln, Lincoln, U.K. He was a Research and Development Engineer with Imperial College London, and Selex Galileo, Ltd., London, after the completion of the Ph.D. degree. Following that, he was a Research Associate with the Department of Computing, Intelligent Behaviour Understanding Group, Imperial College London, He is currently an Associate Editor of the *Image and Vision Computing Journal*. His current research interests include face and object recognition, alignment and tracking, and facial expression analysis.

Maja Pantic (M'98–SM'06-F'12) is a Professor of affective and behavioral computing with the Department of Computing, Imperial College London, London, U.K., and with the Department of Computer Science, University of Twente, Enschede-Noord, The Netherlands.

She was a recipient of various awards for her work on automatic analysis of human behaviors, including the European Research Council Starting Grant Fellowship in 2008 for being in the top 2% of the best young scientists in any research field in Europe, and the British Computer Society Roger Needham Award in 2011 for a distinguished research contribution in computer science within ten years of her Ph.D. She is currently the Editor-in-Chief of the *Image and Vision Computing Journal*, and an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS.