On One-Shot Similarity Kernels: explicit feature maps and properties

Stefanos Zafeiriou[†]
[†]Department of Computing
Imperial College London
s.zafeiriou@imperial.ac.uk

Irene Kotsia[*,†,⋆]
[*]Electronics Laboratory, Department of Physics,
University of Patras, Greece
[⋆]School of Science and Technology,
Middlesex University, London
i.kotsia@mdx.ac.uk

Abstract

Kernels have been a common tool of machine learning and computer vision applications for modeling non-linearities and/or the design of robust[1] similarity measures between objects. Arguably, the class of positive semi-definite (psd) kernels, widely known as Mercer's Kernels, constitutes one of the most well-studied cases. For every psd kernel there exists an associated feature map to an arbitrary dimensional Hilbert space $\mathcal{H}$, the so-called feature space. The main reason behind psd kernels' popularity is the fact that classification/regression techniques (such as Support Vector Machines (SVMs)) and component analysis algorithms (such as Kernel Principal Component Analysis (KPCA)) can be devised in $\mathcal{H}$, without an explicit definition of the feature map, only by using the kernel (the so-called kernel trick). Recently, due to the development of very efficient solutions for large scale linear SVMs and for incremental linear component analysis, the research towards finding feature map approximations for classes of kernels has attracted significant interest. In this paper, we attempt the derivation of explicit feature maps of a recently proposed class of kernels, the so-called one-shot similarity kernels. We show that for this class of kernels either there exists an explicit representation in feature space or the kernel can be expressed in such a form that allows for exact incremental learning. We theoretically explore the properties of these kernels and show how these kernels can be used for the development of robust visual tracking, recognition and deformable fitting algorithms.

1. Introduction

In kernel learning[2] [26], for each positive semi definite (psd) kernel k there exists an associated feature map ϕ to an arbitrary dimensional (in some cases infinite) feature space $\mathcal{H}$. In that case, the kernel learning problem, even though being nonlinear in the original space, becomes linear in the new space $\mathcal{H}$. The explicit form of ϕ is not required to perform all computations, as the so-called kernel trick (i.e., replacing the inner product with the kernel) can be employed.

Recently, the following reverse problem has attracted a lot of attention: Given a kernel k, one should find an efficient and effective approximation of ϕ that successfully replaces the kernel [27, 18, 2, 15, 16, 2]. The motivation behind this was twofold. The first concerned recent developments in learning Support Vector Machines (SVMs), in which it was showed that it is possible to learn a linear SVM in linear time, with respect to the number of training examples [9] (making the applications of SVMs to large scale databases and structural problems feasible). The second concerned the unavailability of both exact (or effective) and efficient incremental versions of Principal Component Analysis (PCA) algorithms with kernels [4, 11, 7] (there exist only for the linear case [14, 23]). Indeed, the most well known incremental Kernel PCA (KPCA) algorithms presented in [11, 7, 4] use only approximations. The first two [11, 7] find an approximate solution using a Hebbian rule. In [4], the authors kernelized an exact algorithm for incremental PCA [14, 23], but, in order to maintain a constant update speed, they constructed a reduced set of expansions of the kernel principal components and of the mean, using pre-images. However, the method in [4] has two main drawbacks: first, the reduced set representation provides only an approximation to the exact solution and second, the extra optimization problem for finding the reduced expansion inevitably increases the complexity of the algorithm.

[1] Robustness may refer to either the presence of outliers and noise or to the robustness to a class of transformations (e.g., translation).

[2] As kernel learning we refer to the general framework of classification, regression and component analysis with kernels.

As mentioned above, a kernel learning problem becomes linear in the feature space $\mathcal{H}$. Hence, when low-dimensional closed forms or effective, efficient and low-dimensional approximations exist for ϕ we can take full advantage of efficient packages for regression and classification [9] but also of exact and low cost incremental PCA algorithms [14, 23]. Such closed or approximated forms are not, in general, easy to find. However, it was recently shown that for some particular classes of kernels such approximations do exist. The main lines of research towards efficient approximation of features map include (a) exploiting particular kernel properties to find the approximation (e.g., in [27, 18] the authors exploited various properties to propose efficient and effective approximations of large families of additive kernels); (b) the application of random sampling on Fourier features [15, 16, 2] (e.g., in [22] methodologies have been proposed for encoding stationary kernels by randomly sampling their Fourier features); (c) the application of the so-called Nystrom method, which is a data-dependent methodology that requires training [29, 28, 21]. Even though the above methods provide useful and general methodologies that are applicable to many kernels, their disadvantage is that they provide approximate solutions.

In this paper, we study a recently proposed kernel, the so-called one-shot similarity kernel, which was shown to be particularly useful for the recently introduced similarity problems (face and action similarity [32, 30, 12]). In particular, we show that (1) a special form of the kernel has a closed form feature map and (2) the general kernel can be written in a form which allows for efficient incremental solutions. Hence, the proposed form of the one-shot similarity kernel makes it suitable for incremental PCA, which is particular useful for visual tracking [23]. Summarizing, the contributions of this paper are:

- We study the recently proposed class of one-shot similarity kernels and show that there exist closed form solutions that can be acquired after simple data normalization. For this case we show that (1) the use of one-shot similarity kernel with SVMs can be re-interpreted as a margin maximization and (2) the one-shot similarity kernels can be used with the recently introduced SVM packages which can train linear SVMs in linear time (i.e., making them suitable for large datasets).

- We show that the proposed one-shot similarity kernel can be formulated in a form which allows for incremental Principal Component Analysis.

- We apply the one-shot similarity kernel to object tracking where state-of-the-art results are achieved.

2. The One-Shot Similarity Kernel

In this section, we will define the one-shot and multiple-shot similarity kernels, having as an example the recently introduced face similarity problem in the wild [30, 12, 31, 32]. Face similarity is conceptually different to the standard face recognition, in which the algorithm, given a test facial image, should find the most similar face (or the k-most similar faces) from a pre-defined dataset (corresponding to the same identity). Indeed, face similarity tries to determine whether two given facial images belong to the same face or not. Furthermore, there is a subtle, yet crucial difference between the face similarity and verification problems [32, 8, 34, 35]. In face verification the identity being claimed is known, hence person specific models can be learned and used. This is not the case with face similarity, as such models can not be used or trained (the interested reader can refer to [8, 32] and in the references within for more details regarding the face similarity problem).

In order to construct the one-shot similarity kernel, background samples are required. The term background samples $\mathcal{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$ corresponds to samples that do not belong to the classes being learned and can be in the form of either feature vectors or vectors of scores [32]. As their labeling is not required, their collection is easy. For example, in face recognition, as background samples we can consider a set of facial images that do not belong to the list of faces of the system (very similar to the so-called world model in the face verification problem).

Let us assume two vectors $\mathbf{x}$ and $\mathbf{y} \in \Re^d$. Their one-shot similarity score is computed by considering the set of background samples $\mathcal{A}$ with cardinality $N_{\mathcal{A}}$, which contains samples not belonging to the same class as neither $\mathbf{x}$ nor $\mathbf{y}$ but otherwise not-labeled [32]. The one-shot similarity score in the Fisher's Linear Discriminant Analysis (FLDA) framework can be described as follows.

Let the covariance matrix of the set $\mathcal{A}$ be defined as:

$$\mathbf{S} = \frac{1}{N_{\mathcal{A}}} \sum_{i=1}^{N_{\mathcal{A}}} (\mathbf{a}_i - \mathbf{m}_{\mathcal{A}})(\mathbf{a}_i - \mathbf{m}_{\mathcal{A}})^T \tag{1}$$

where $\mathbf{m}_{\mathcal{A}} = \frac{1}{N_{\mathcal{A}}} \sum_{i=1}^{N_{\mathcal{A}}} \mathbf{a}_i$. Then the one-shot similarity kernel measures the similarity between $\mathbf{x}$ and $\mathbf{y}$ via the background samples $\mathcal{A}$ in a FLDA manner as:

$$k_{\mathcal{A}}(\mathbf{x}, \mathbf{y}) = \frac{(\mathbf{x} - \mathbf{m}_{\mathcal{A}})^T \mathbf{S}^{-1} (\mathbf{y} - \frac{\mathbf{x} + \mathbf{m}_{\mathcal{A}}}{2})}{||\mathbf{S}^{-1}(\mathbf{x} - \mathbf{m}_{\mathcal{A}})||} + \frac{(\mathbf{y} - \mathbf{m}_{\mathcal{A}})^T \mathbf{S}^{-1} (\mathbf{x} - \frac{\mathbf{y} + \mathbf{m}_{\mathcal{A}}}{2})}{||\mathbf{S}^{-1}(\mathbf{y} - \mathbf{m}_{\mathcal{A}})||} \tag{2}$$

or the non-normalized kernel:

$$k_{\mathcal{A}}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{m}_{\mathcal{A}})^T \mathbf{S}^{-1} (\mathbf{y} - \frac{\mathbf{x} + \mathbf{m}_{\mathcal{A}}}{2}) + (\mathbf{y} - \mathbf{m}_{\mathcal{A}})^T \mathbf{S}^{-1} (\mathbf{x} - \frac{\mathbf{y} + \mathbf{m}_{\mathcal{A}}}{2}). \tag{3}$$

The above kernel in (3), as proven in [30], is a psd kernel (for further details regarding the one-shot similarity kernel the interested reader may refer to [30, 12, 31, 32]).

In the following we will show how the above kernel (3): (1) has a very simple closed form for the case in which the samples $\mathbf{x}$ and background samples $\mathcal{A}$ are normalized and (2) can be written in a very convenient form that allows for both effective and efficient incremental component analysis.

3. Properties of the Kernel

Let us assume $\tilde{\mathbf{x}} = \mathbf{S}^{-\frac{1}{2}}\mathbf{x}$, $\tilde{\mathbf{m}}_{\mathcal{A}} = \mathbf{S}^{-\frac{1}{2}}\mathbf{m}_{\mathcal{A}}$ and $\tilde{\mathbf{y}} = \mathbf{S}^{-\frac{1}{2}}\mathbf{y}$. Then the kernel in (3) can be written as:

$$
\begin{aligned}
&k_{\mathcal{A}}(\mathbf{x},\mathbf{y}) = \\
&(\tilde{\mathbf{x}} - \tilde{\mathbf{m}}_{\mathcal{A}})^T(\tilde{\mathbf{y}} - \frac{\tilde{\mathbf{x}} + \tilde{\mathbf{m}}_{\mathcal{A}}}{2}) + (\tilde{\mathbf{y}} - \tilde{\mathbf{m}}_{\mathcal{A}})^T(\tilde{\mathbf{x}} - \frac{\tilde{\mathbf{y}} + \tilde{\mathbf{m}}_{\mathcal{A}}}{2}) \\
&= \tilde{\mathbf{x}}^T\tilde{\mathbf{y}} - \tilde{\mathbf{m}}_{\mathcal{A}}^T\tilde{\mathbf{y}} - \frac{\tilde{\mathbf{x}}^T\tilde{\mathbf{x}}}{2} - \frac{\tilde{\mathbf{x}}^T\tilde{\mathbf{m}}_{\mathcal{A}}}{2} + \frac{\tilde{\mathbf{m}}_{\mathcal{A}}^T\tilde{\mathbf{x}}}{2} + \frac{\tilde{\mathbf{m}}_{\mathcal{A}}^T\tilde{\mathbf{m}}_{\mathcal{A}}}{2} \\
&+ \tilde{\mathbf{y}}^T\tilde{\mathbf{x}} - \frac{\tilde{\mathbf{y}}^T\tilde{\mathbf{y}}}{2} - \frac{\tilde{\mathbf{y}}^T\tilde{\mathbf{m}}_{\mathcal{A}}}{2} - \tilde{\mathbf{m}}_{\mathcal{A}}^T\tilde{\mathbf{x}} + \frac{\tilde{\mathbf{m}}_{\mathcal{A}}^T\tilde{\mathbf{y}}}{2} + \frac{\tilde{\mathbf{m}}_{\mathcal{A}}^T\tilde{\mathbf{m}}_{\mathcal{A}}}{2} \\
&= 2(\tilde{\mathbf{x}} - \frac{\tilde{\mathbf{m}}_{\mathcal{A}}}{2})^T(\tilde{\mathbf{y}} - \frac{\tilde{\mathbf{m}}_{\mathcal{A}}}{2}) - \frac{\tilde{\mathbf{x}}^T\tilde{\mathbf{x}}}{2} - \frac{\tilde{\mathbf{y}}^T\tilde{\mathbf{y}}}{2} \\
&= 2(\mathbf{x} - \frac{\mathbf{m}_{\mathcal{A}}}{2})^T\mathbf{S}^{-1}(\mathbf{y} - \frac{\mathbf{m}_{\mathcal{A}}}{2}) - \frac{\mathbf{x}^T\mathbf{S}^{-1}\mathbf{x}}{2} - \frac{\mathbf{y}^T\mathbf{S}^{-1}\mathbf{y}}{2}.
\end{aligned} \tag{4}
$$

The kernel k can thus take the following functional form:

$$k(\mathbf{x},\mathbf{y}) = \mathbf{f}(\mathbf{x})^T\mathbf{g}(\mathbf{y}) \tag{5}$$

where

$$\mathbf{f}(\mathbf{u}) = \begin{bmatrix} \sqrt{2}\mathbf{S}^{-\frac{1}{2}}(\mathbf{u} - \frac{\tilde{\mathbf{m}}_{\mathcal{A}}}{2}) \\ -\frac{1}{\sqrt{2}}\left(\mathbf{S}^{-\frac{1}{2}}\mathbf{u}\right) \odot \left(\mathbf{S}^{-\frac{1}{2}}\mathbf{u}\right) \\ \mathbf{1} \end{bmatrix} \tag{6}$$

and

$$\mathbf{g}(\mathbf{u}) = \begin{bmatrix} \sqrt{2}\mathbf{S}^{-\frac{1}{2}}(\mathbf{u} - \frac{\tilde{\mathbf{m}}_{\mathcal{A}}}{2}) \\ \mathbf{1} \\ -\frac{1}{\sqrt{2}}\left(\mathbf{S}^{-\frac{1}{2}}\mathbf{u}\right) \odot \left(\mathbf{S}^{-\frac{1}{2}}\mathbf{u}\right). \end{bmatrix} \tag{7}$$

and $\odot$ is the Hadamard product of vectors (i.e., $\mathbf{a} \odot \mathbf{b} = [a_i b_i]$). A nice property of the kernel is:

$$k(\mathbf{x},\mathbf{y}) = \mathbf{f}(\mathbf{x})^T\mathbf{g}(\mathbf{y}) = \mathbf{f}(\mathbf{y})^T\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{y})^T\mathbf{f}(\mathbf{x}). \tag{8}$$

In the next section we are going to exploit this property to formulate an exact and incremental Principal Component Analysis (iPCA). Finally, it is important to note here that, even though the mappings $\mathbf{f}(.)$ and $\mathbf{g}(.)$ are known (and of course $\mathbf{f}(.) \neq \mathbf{g}(.)$), the mapping $\phi(.)$ associated to the kernel k is not known and neither can be explicitly defined, unless $\mathbf{x}^T\mathbf{S}^{-1}\mathbf{x}$ and $\mathbf{y}^T\mathbf{S}^{-1}\mathbf{y}$ are known. In the following, we will assume that the training data are previously normalized, such that $\mathbf{x}^T\mathbf{S}^{-1}\mathbf{x}$ and $\mathbf{y}^T\mathbf{S}^{-1}\mathbf{y}$ are constants.

3.1. A Special Case of the one-shot similarity kernel

Assuming that all data are normalized such that $||\mathbf{x}||_{\mathbf{S}}^2 = \tilde{\mathbf{x}}^T\tilde{\mathbf{x}} = \tilde{\mathbf{y}}^T\tilde{\mathbf{y}} = 1$ (i.e., $\mathbf{x}^T\mathbf{S}^{-1}\mathbf{x} = 1$ and $\mathbf{y}^T\mathbf{S}^{-1}\mathbf{y} = 1$), then the kernel, after removing the constant terms and the global translation by $\frac{\tilde{\mathbf{m}}_{\mathcal{A}}}{2}$, can be written as the simple dot product:

$$k_{\mathcal{A}}(\tilde{\mathbf{x}},\tilde{\mathbf{y}}) = \tilde{\mathbf{x}}^T\tilde{\mathbf{y}}. \tag{9}$$

Hence, the closed feature map that can be used has the following closed form:

$$\phi(\mathbf{x}) = \mathbf{S}^{-\frac{1}{2}}\mathbf{x}. \tag{10}$$

We will now study the interpretation of the application of this kernel within the SVMs framework.

Let a set of labeled samples $\mathbf{x}_1,\ldots,\mathbf{x}_n$, with an accompanying set of labels $l_1,\ldots,l_n$, $l_i \in \{-1,1\}$ (normalized such that $||\mathbf{x}_i||_{\mathbf{S}} = 1$) and a set of background samples $\mathcal{A}$ with their corresponding covariance matrix $\mathbf{S}$. SVMs aim at finding a hyperplane of the form $\mathbf{w}^T\mathbf{x} + b$ by maximizing the margin of the data subject to data separability constraints. Typically, $\mathbf{w}$ and b are found by solving the Wolf dual problem where in the case of the kernel (3) can be written as:

$$\max_{0\leq\alpha_i\leq C}\ \boldsymbol{\alpha}^T\mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^T\mathbf{K}_s\boldsymbol{\alpha},\ \text{s.t.}\ \sum_{i=1}^{l} l_i\alpha_i = 0 \tag{11}$$

where $\mathbf{K}_s = [l_i l_j k_{\mathcal{A}}(\mathbf{x}_i,\mathbf{x}_j)]$ and $\mathbf{w} = \sum_{i=1}^{n} l_i\alpha_i\mathbf{x}_i$. The above dual problem is equivalent to the following primal problem:

$$\min_{\mathbf{w}}\ \frac{1}{2}\mathbf{w}^T\mathbf{S}\mathbf{w} + C\sum_{i=1}^{n}\xi_i,\ \text{s.t.}\ l_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i. \tag{12}$$

Thus, when the one-shot similarity kernel is used with SVMs, it attempts to maximize a squared Mahalanobis type distance margin, which is inversely proportional to $\mathbf{w}^T\mathbf{S}\mathbf{w}$. Hence, the one-shot similarity kernel can be interpreted as a type of margin being maximized within the linear SVM framework. In case the matrix $\mathbf{S}$ is singular or in non-linear case where the one-shot similarity kernel is used in a features space (i.e., a kernel is used in the SVM problem (12)), solutions can be provided by using the tools in ([36, 13]).

Since the kernel has a closed form it can be directly used with the recently proposed linear SVMs which can be trained in linear time with regards to the number of training samples and solve the following reformulated optimization

problem [3]:

$$\min_{\mathbf{w},\xi} \ \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\xi \text{ s.t. } \forall \mathbf{c} \in \{0,1\}^n \quad (13)$$
$$\frac{1}{n}\mathbf{w}^T \sum_{i=1}^{n} c_i l_i \mathbf{S}^{-\frac{1}{2}}(\mathbf{x} - \mathbf{m}_{\mathcal{A}}) \geq \frac{1}{n}\sum_{i=1}^{n} c_i - \xi.$$

It is worth noting here that the functional form of the similarity kernel in (3) does not allow the use of fast cutting plane algorithm for solving (13), as proposed in [10].

4. Exact Incremental Component Analysis using the one-shot similarity kernel

In this section, we will show how the property in (8) can be harnessed in order to define a special version of KPCA. The proposed KPCA, contrary to the general incremental KPCA approaches [4], does not require the computation of pre-images. The following analysis is similar to the one presented in [17] but now refers to strictly psd kernels.

Let $\mathbf{X}_\phi = \left[\begin{array}{ccc} \phi(\mathbf{x}_1) & \cdots & \phi(\mathbf{x}_N) \end{array} \right]$ be the matrix of N known samples in Hilbert space defined by the kernel (3) (for simplicity we assume zero mean [4]). We define the matrices $\mathbf{X}_f = \left[\begin{array}{ccc} \mathbf{f}(\mathbf{x}_1) & \cdots & \mathbf{f}(\mathbf{x}_N) \end{array} \right]$ and $\mathbf{X}_g = \left[\begin{array}{ccc} \mathbf{g}(\mathbf{x}_1) & \cdots & \mathbf{g}(\mathbf{x}_N) \end{array} \right]$. For the one-shot similarity kernel, even though $\mathbf{X}_\phi$ cannot be explicitly defined, $\mathbf{X}_g$ and $\mathbf{X}_f$ can. The fact that we have explicit mappings for $\mathbf{X}_g$ and $\mathbf{X}_f$ makes feasible the computation of incremental PCA without the use of pre-images.

In KPCA we want to find a set of projections in the feature space such that:

$$\mathbf{U}_\phi^o = \max_{\mathbf{U}_\phi} \text{tr}[\mathbf{U}_\phi^T \mathbf{X}_\phi \mathbf{X}_\phi^T \mathbf{U}_\phi] \quad (14)$$
$$\text{s.t.} \mathbf{U}_\phi^T \mathbf{U}_\phi = \mathbf{I}. \quad (15)$$

Unfortunately, $\mathbf{U}_\phi$ cannot be computed directly as in the majority of cases the eigen-analysis of $\mathbf{X}_\phi \mathbf{X}_\phi^T$ is computationally expensive or ϕ is not known. In [25] it was shown that we can instead perform eigen-analysis on the Gram matrix $\mathbf{K} = \mathbf{X}_\phi^T \mathbf{X}_\phi$. Due to this property we have the following eigen-decomposition:

$$\mathbf{X}_\phi^T \mathbf{X}_\phi = \mathbf{X}_f^T \mathbf{X}_g = \mathbf{X}_g^T \mathbf{X}_f = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T. \quad (16)$$

The projection bases $\mathbf{U}_\phi$ of KPCA are given by $\mathbf{U}_\phi = \mathbf{X}_\phi \mathbf{V}\mathbf{\Lambda}^{-\frac{1}{2}}$ (which cannot be explicitly computed). We define $\mathbf{U}_f \triangleq \mathbf{X}_f \mathbf{V}\mathbf{\Lambda}^{-\frac{1}{2}}$ and $\mathbf{U}_g \triangleq \mathbf{X}_g \mathbf{V}\mathbf{\Lambda}^{-\frac{1}{2}}$. Hence, we have explicit decompositions for $\mathbf{X}_f$ and $\mathbf{X}_g$ as $\mathbf{X}_f = \mathbf{U}_f \mathbf{\Sigma}\mathbf{V}^T$ and $\mathbf{X}_g = \mathbf{U}_g \mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{\Sigma} = \mathbf{\Lambda}^{\frac{1}{2}}$. Additionally, using the kernel properties (8) the following properties hold:

$$\mathbf{U}_\phi^T \phi(\mathbf{x}) = \mathbf{U}_f^T \mathbf{g}(\mathbf{x}) = \mathbf{U}_g^T \mathbf{f}(\mathbf{x}) \quad (17)$$

and also $\mathbf{U}_f$ and $\mathbf{U}_g$ are mutually orthogonal

$$\begin{aligned} \mathbf{U}_f^T \mathbf{U}_g &= \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{V}^T \mathbf{X}_f^T \mathbf{X}_g \mathbf{V}\mathbf{\Lambda}^{-\frac{1}{2}} \\ &= \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{V}^T \mathbf{X}_\phi^T \mathbf{X}_\phi \mathbf{V}\mathbf{\Lambda}^{-\frac{1}{2}} \\ &= \mathbf{U}_\phi^T \mathbf{U}_\phi = \mathbf{I}. \end{aligned} \quad (18)$$

We proceed with showing that by using the explicit definition of $\mathbf{U}_f$ and $\mathbf{U}_g$ we can define an incremental KPCA without the need of pre-images. Let us assume two initial subspaces $\mathbf{U}_f$ and $\mathbf{U}_g$ and a number of incoming data $\tilde{\mathbf{X}} = \left[\begin{array}{ccc} \mathbf{x}_{N+1} & \cdots & \mathbf{x}_{N+M} \end{array} \right]$. Incremental KPCA aims at updating the subspaces $\mathbf{U}_f$ and $\mathbf{U}_g$ without computing KPCA from scratch.

$\tilde{\mathbf{X}}_\phi = [\phi(\mathbf{x}_{N+1}) \ \cdots \ \phi(\mathbf{x}_{N+M})]$ is the data matrix of the new data in the feature space. For these data we define the explicit maps $\tilde{\mathbf{X}}_f = [\mathbf{f}(\mathbf{x}_{N+1}) \ \cdots \ \mathbf{f}(\mathbf{x}_{N+M})]$ and $\tilde{\mathbf{X}}_g = [\mathbf{g}(\mathbf{x}_{N+1}) \ \cdots \ \mathbf{g}(\mathbf{x}_{N+M})]$. Finally, we denote the combined sample matrix by $\left[\begin{array}{cc} \mathbf{X}_\phi & \tilde{\mathbf{X}}_\phi \end{array} \right]$, where $\mathbf{X}_\phi$ are the previously available data in $\mathcal{H}$. The combined matrix is equivalent to [4]:

$$\left[\begin{array}{cc} \mathbf{U}_\phi \mathbf{\Sigma}\mathbf{V}^T & \mathbf{U}_\phi \mathbf{U}_\phi^T \tilde{\mathbf{X}}_\phi + \mathbf{Q}_\phi \mathbf{R}_\phi \end{array} \right] \quad (19)$$

where $\mathbf{Q}_\phi$ is an orthogonal matrix and $\mathbf{Q}_\phi \mathbf{R}_\phi = \mathbf{H}_\phi$. $\mathbf{H}_\phi = \tilde{\mathbf{X}}_\phi - \mathbf{U}_\phi \mathbf{U}_\phi^T \tilde{\mathbf{X}}_\phi$ is the complementary to the $\mathbf{U}_\phi$ subspace,. We obtain $\mathbf{Q}_\phi = \mathbf{H}_\phi \mathbf{\Omega}\mathbf{\Delta}^{-\frac{1}{2}}$ and $\mathbf{R}_\phi = \mathbf{\Delta}^{\frac{1}{2}}\mathbf{\Omega}^T$ by the eigendecomposition of $\mathbf{H}_\phi^T \mathbf{H}_\phi = \mathbf{\Omega}\mathbf{\Delta}\mathbf{\Omega}^T$. We define $\mathbf{H}_f \triangleq \tilde{\mathbf{X}}_f - \mathbf{U}_f \mathbf{U}_g^T \tilde{\mathbf{X}}_f$ and $\mathbf{H}_g \triangleq \tilde{\mathbf{X}}_g - \mathbf{U}_g \mathbf{U}_f^T \tilde{\mathbf{X}}_g$ and compute the eigendecomposition of $\mathbf{H}_f^T \mathbf{H}_g$ to avoid the computation of the projection of $\tilde{\mathbf{X}}_\phi$ onto $\mathbf{U}_\phi$ as:

$$\begin{aligned} \mathbf{H}_f^T \mathbf{H}_g &= (\tilde{\mathbf{X}}_f^T - \tilde{\mathbf{X}}_f^T \mathbf{U}_g \mathbf{U}_f^T)(\tilde{\mathbf{X}}_g - \mathbf{U}_g \mathbf{U}_f^T \tilde{\mathbf{X}}_g) \\ &= (\tilde{\mathbf{X}}_\phi - \mathbf{U}_\phi \mathbf{U}_\phi^T \tilde{\mathbf{X}}_\phi)^T (\tilde{\mathbf{X}}_\phi - \mathbf{U}_\phi \mathbf{U}_\phi^T \tilde{\mathbf{X}}_\phi) \\ &= \mathbf{H}_\phi^T \mathbf{H}_\phi = \mathbf{\Omega}\mathbf{\Delta}\mathbf{\Omega}^T. \end{aligned} \quad (20)$$

The matrix in (19) can be rewritten as

$$\left[\begin{array}{cc} \mathbf{U}_\phi & \mathbf{Q}_\phi \end{array} \right] \mathbf{L}_\phi \left[\begin{array}{cc} \mathbf{V}_\phi^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{array} \right] \quad (21)$$

where $\mathbf{L}_\phi = \left[\begin{array}{cc} \mathbf{\Sigma}_\phi & \mathbf{U}_\phi^T \tilde{\mathbf{X}}_\phi \\ \mathbf{0} & \mathbf{R}_\phi \end{array} \right]$. The SVD of $\left[\begin{array}{cc} \mathbf{X}_\phi & \tilde{\mathbf{X}}_\phi \end{array} \right]$ is then given by:

$$\left[\left[\begin{array}{cc} \mathbf{U}_\phi & \mathbf{Q}_\phi \end{array} \right] \tilde{\mathbf{U}}_\phi \right] \left[\tilde{\mathbf{\Sigma}}_\phi \right] \left[\tilde{\mathbf{V}}_\phi \left[\begin{array}{cc} \mathbf{V}_\phi^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{array} \right] \right] \quad (22)$$

[3] The optimization problem (13) theoretically provides the same solution as the quadratic program (11) but can be solved efficiently using a cutting plane algorithm in linear time.

[4] Centering in the feature space is straightforward by centering the kernel matrix [25].

where $\mathbf{L}_\phi \overset{svd}{=} \tilde{\mathbf{U}}_\phi \tilde{\mathbf{\Sigma}}_\phi \tilde{\mathbf{V}}_\phi^T$. Thus, we only need to compute the SVD of $\mathbf{L}_\phi$ for the incremental update of our eigenspace, $\mathbf{U}'_\phi = \begin{bmatrix} \mathbf{U}_\phi & \mathbf{L}_\phi \end{bmatrix} \tilde{\mathbf{U}}_\phi$ and $\mathbf{\Sigma}'_\phi = \tilde{\mathbf{\Sigma}}_\phi$. As $\mathbf{U}_\phi$ and $\mathbf{H}_\phi$ are not directly given by our KPCA (and actually they are not essential), we define $\mathbf{Q}_f \triangleq \mathbf{H}_f \mathbf{\Omega} \mathbf{\Delta}^{-\frac{1}{2}}$ and $\mathbf{Q}_g \triangleq \mathbf{H}_g \mathbf{\Omega} \mathbf{\Delta}^{-\frac{1}{2}}$, and set $\mathbf{U}'_f = \begin{bmatrix} \mathbf{U}_f & \mathbf{Q}_f \end{bmatrix} \tilde{\mathbf{U}}_\phi$ and $\mathbf{U}'_g = \begin{bmatrix} \mathbf{U}_g & \mathbf{Q}_g \end{bmatrix} \tilde{\mathbf{U}}_\phi$. Note that this satisfies (17) and (18). Algorithm 1 summarizes the proposed incremental update. Due to our direct approach to KPCA, the storage requirements for the incremental update is of fixed complexity (*e.g.* $\mathcal{O}(4d(p+M))$ for our kernel, where p is the number of eigen-components we update. The complexity of the update is also fixed for our kernel (*e.g.* in $\mathcal{O}(2dM^2)$, similarly to [23]. Finally, in contrast to the incremental version of KPCA proposed in [4], the extra optimization step required to find the pre-images is not necessary. Therefore, the proposed method is not only faster but also exact.

One of the main applications of iPCA is object tracking [23], in which the object subspace is adaptively learned and online updated. In this paper we combine the proposed kernel with the tracking framework proposed in [23], but instead of PCA we use the KPCA with the proposed kernel. In brief, in [23], in order to find the parameters of the motion a particle filter framework is used [20]. At each frame a number of particles (containing motion parameters) are drawn. The particle chosen is the one corresponding to an image which can be best reconstructed within a subspace of choice (in our case, our kernel subspace). The reconstruction is measured by:

$$\begin{aligned} D \quad & (\mathbf{x}_i) = (\phi(\mathbf{x}_i) - \mathbf{U}_\phi \mathbf{U}_\phi^T)^T (\phi(\mathbf{x}_i) - \mathbf{U}_\phi \mathbf{U}_\phi^T) \\ & = (\mathbf{f}(\mathbf{x}_i) - \mathbf{U}_f \mathbf{U}_g^T \mathbf{g}(\mathbf{x}_i))^T (\mathbf{g}(\mathbf{x}_i) - \mathbf{U}_g \mathbf{U}_f^T \mathbf{f}(\mathbf{x}_i)) \end{aligned} \tag{23}$$

which in our case (using the proposed kernel) can be defined using only $\mathbf{U}_f$ and $\mathbf{U}_g$, whose explicit updates are available from the proposed KCPA. In the tracking framework the set of background samples $\mathcal{A}$ needed can be images of objects other than the one we wish to sample. In our case we adaptively learn $\mathcal{A}$, updating it in the first 20 frames by keeping around 100-200 images that correspond to the lowest, in probability, particles in every frame. In case $\mathbf{S}$ is singular we compute as $\mathbf{S}^{-1} = \mathbf{U}_S \mathbf{\Lambda}_S^{-1} \mathbf{U}_S^T$ where $\mathbf{\Lambda}_S$ is the strictly positive spectrum of $\mathbf{S}$.

Algorithm 1 INCREMENTAL UPDATE OF KPCA WITH THE ONE-SHOT SIMILARITY KERNEL

Require: The previous eigenspaces $\mathbf{U}_f$, $\mathbf{U}_g$ and $\mathbf{\Sigma}_\phi$, and the number of previous samples N, the set of M new samples $\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{x}_{N+1} & \cdots & \mathbf{x}_{N+M} \end{bmatrix} \in \Re^{d \times M}$ and the two mappings $\mathbf{f}(.)$ and $\mathbf{g}(.)$.

Ensure: The updated eigenspaces $\mathbf{U}'_f$, $\mathbf{U}'_g$ and $\mathbf{\Sigma}'_\phi$.

1: Calculate the mappings, $\tilde{\mathbf{X}}_f$ and $\tilde{\mathbf{X}}_g$, of $\tilde{\mathbf{X}}$.
2: Find $\mathbf{H}_f = \tilde{\mathbf{X}}_f - \mathbf{U}_f \mathbf{U}_g^T \tilde{\mathbf{X}}_f$ and $\mathbf{H}_g = \tilde{\mathbf{X}}_g - \mathbf{U}_g \mathbf{U}_f^T \tilde{\mathbf{X}}_g$.
3: Compute $\mathbf{H}_f^T \mathbf{H}_g = \mathbf{H}_\phi^T \mathbf{H}_\phi = \mathbf{\Omega} \mathbf{\Delta} \mathbf{\Omega}^T$ and set $\mathbf{R}_\phi = \mathbf{\Delta}^{\frac{1}{2}} \mathbf{\Omega}^T$, $\mathbf{Q}_f = \mathbf{H}_f \mathbf{\Omega} \mathbf{\Delta}^{-\frac{1}{2}}$ and $\mathbf{Q}_g = \mathbf{H}_g \mathbf{\Omega} \mathbf{\Delta}^{-\frac{1}{2}}$.
4: Form $\mathbf{L}_\phi = \begin{bmatrix} \mathbf{\Sigma}_\phi & \mathbf{U}_g^T \tilde{\mathbf{X}}_f \\ \mathbf{0} & \mathbf{R}_\phi \end{bmatrix}$ and compute $\mathbf{L}_\phi \overset{svd}{=} \tilde{\mathbf{U}}_\phi \tilde{\mathbf{\Sigma}}_\phi \tilde{\mathbf{V}}_\phi^T$.
5: Set $\mathbf{U}'_f = \begin{bmatrix} \mathbf{U}_f & \mathbf{Q}_f \end{bmatrix} \tilde{\mathbf{U}}_\phi$, $\mathbf{U}'_g = \begin{bmatrix} \mathbf{U}_g & \mathbf{Q}_g \end{bmatrix} \tilde{\mathbf{U}}_\phi$.
6: Obtain the p-reduced set of $\mathbf{U}_f$ and $\mathbf{U}_g$ *via* the p largest eigenvalue magnitudes in $\tilde{\mathbf{\Sigma}}_\phi$.

5. Experimental Results

For our experiments, we evaluated the proposed kernel in a number of applications including face recognition, object tracking and deformable model fitting. For face recognition we used a similar framework to [32]. We show that the proposed formulation can obtain similar results but in linear training time. Furthermore, we applied the one-shot similarity kernel to object tracking by combining the one-shot similarity kernel with the proposed incremental subspace learning framework. We show that by exploiting the functional form of the one-shot similarity kernel state-of-the-art tracking results can be produced. Finally, we combined the one-shot similarity kernel- SVMs with the recently introduced discriminative fitting algorithms, such as the Constrained Local Models (CLMs) [24] and we report state-of-the-art fitting results.

5.1. Face Recognition

The usefulness of the one-shot similarity kernel for face verification has been shown in [32] , using the LFW database [8], hence we do not repeat these experiments. We did perform multi-identity face classification in the LFW image set to show the gain in computational time by using the proposed formulation of the one-shot similarity kernel. In more detail, we tried to perform the same face recognition experiments as in [32, 33]. We selected a subset of the LFW database with subjects having at least four images. This subset contains 610 subjects with 6733 images. As in [32, 33], we fused various features and measured the face recognition performance by varying the number of probes (from 5, 10, 20 and 50 subjects) and performing 20 random repetitions per experiment (for more details regarding the features the interested reader can refer to [32, 33]).

We used the one-vs-all linear SVM proposed in [6] with the original form of the one-shot similarity kernel in (3). We also used the proposed form of the kernel (9) with a fast implementation of one-vs-all linear SVM in [10], for comparison reasons. This implementation can be used only for the case of linear kernels (or, as in our case, only when $\phi(\mathbf{x})$ is known). As in [32] for the definition of the negative set $\mathcal{A}$ a set of 1000 images were selected at random from the remaining individuals having only one image.

The mean classification rate and the variance for the random 20 runs is summarized in Table 1. The original one-

Table 1. Mean Classification Accuracy and Variance performance. Columns represent the number of subjects.

	5	10	20	50
OSK	0.742 ± 0.1621	0.732 ± 0.987	0.6934 ± 0.921	0.5728 ± 0.0672
F-OSK	0.738 ± 0.1896	0.728 ± 1.072	0.7001 ± 0.945	0.5745 ± 0.0624

shot similarity kernel and the closed form of the one-shot similarity kernel given in (9)) are denoted as OSK and F-OSK, respectively. As we can see, the proposed closed form solution of the one-shot similarity kernel in (9) produces similar results as the original form of the one-shot similarity kernel, but in at least one order of magnitude less time ($O(n)$ over $O(n^2)$ where n are the training samples). Similar gain in computational time was recently reported in [27] using approximations of various additive kernels.

5.2. Object Tracking

We evaluated the performance of our subspace learning algorithms for the application of appearance-based face tracking. The appearance-based approach to tracking has been one of the de facto choices for tracking objects in image sequences. As discussed, the proposed kernel subspace-based tracking algorithm is closely related to the incremental visual tracker in [23] (abbreviated as IVT). As such, our tracker can deal with drastic appearance changes, does not require offline training, continually updates a compact object representation and uses the Condensation algorithm to robustly estimate the object's location [23].

We conducted experiments in order to show (1) that the use of the one-shot similarity kernel and background samples help eigen-tracking and (2) that the proposed formulation of the one-shot similarity kernel given in (8) combined with the proposed incremental PCA is not only faster but also more accurate than using the one-shot similarity kernel with incremental KPCA [4]. Furthermore, we compared with two publicly available state-of-the-art trackers, namely the ℓ_1 tracker proposed in [19] and the Multiple Instance Learning (MIL) tracker in [1]. We evaluated the performance of all methods on nine very popular video sequences, $S_i, i = 1, \ldots, 9$ (subsets of which are used in [23], [19], and [5, 1]). The videos contain drastic changes of the target's appearance, including pose variation, occlusions, and non-uniform illumination. The un-optimized MATLAB code using the proposed iPCA tracks 7-8 frames/sec, while with the original form tracks 1 frame/15 secs (it needs an extra optimization problem for the pre-images [4]).

S_1 is provided along with seven annotated points which indicate the ground truth. We also annotated 3–7 fiducial points for the remaining sequences. Our quantitative performance evaluation is based on the root mean square (RMS) errors between the true and the estimated locations of these points [23]. In our experiments, all trackers use the same motion models (an affine motion model) with a fixed number of drawn particles (800 particles). The parameters of all trackers have been learned in a different set of videos than the tested ones and were kept fixed for all tested videos. The Mean RMS is summarized in Table 2 (the proposed tracker is under the abbreviation OSK-IVT, while the one-shot similarity kernel using the online kernel learning technique with pre-images [4] is abbreviated as OSK-IVT-K). As can be seen, by exploiting the functional form (8) of the proposed kernel we obtain an adaptive tracking algorithm with the same complexity of IVT while producing state-of-the-art results. We have conducted a statistical test and we can verify that improvements in $S_1, S_2, S_3, S_6 \& S_8$ are statistically significant in a 99% confidence interval.

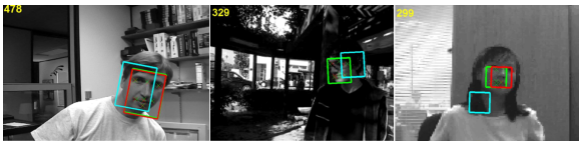

Figure 1. Examples of tracking results (ground truth, IVT and OSK-IVT bounding boxes are colored red, blue and green).

Table 2. Mean RMS error for general tracking. "(lost)" indicates sequences in which the tracker clearly does not follow the target throughout the entire sequence.

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
IVT	8.13	(lost)	4.13	13.14	(lost)	27.79	2.02	24.48
OSK-IVT-K	(lost)	(lost)	(lost)	(lost)	(lost)	(lost)	(lost)	(lost)
L1	(lost)	(lost)	**2.87**	(lost)	12.94	(lost)	1.67	39.15
MIL	51.36	(lost)	13.61	17.78	38.19	(lost)	4.14	40.80
OSK-IVT	**5.11**	**3.47**	3.16	**10.56**	(lost)	**8.89**	1.82	**11.9**

Some indicative examples in which the proposed tracker (using the proposed kernel) outperforms the IVT kernel are shown in Fig. 1. In these images, the bounding box obtained from the ground truth, the IVT bounding box and the OSK-IVT bounding box are colored red, blue and green, respectively. As we can see, even in the case that there is no ground truth available, due to missing points (as the profile frame in the middle image), the proposed tracker provides a more accurate estimation of the bounding box position.

5.3. Deformable Model Fitting

The state-of-the-art algorithms for deformable model fitting are the recently introduced non-parametric CLMs using non-parametric density estimation. CLM is a part based deformable model which comprises of a shape model $\mathcal{S}$ (usually a 3D point distributional model) and a set of detectors $\mathcal{D}$ of the various facial parts (each part corresponds to a fiducial point of $\mathcal{S}$). More formally, $\mathcal{D}$ could be a set of linear classifiers of n parts of the face and can be represented as $\mathcal{D} = \{\mathbf{w}_i, b_i\}_{i=1}^n$, where $\mathbf{w}_i, b_i$ is the linear detector (i.e., SVM) for the i-th part of the face (e.g., eye-corner detector). These detectors are used to define probability maps for the i-th part and for a given location x of an image I being

correctly located ($l_i = 1$) as:

$$p(l_i = 1|\mathbf{x}, I) = \frac{1}{1 + \exp(l_i(\mathbf{w}_i^T \mathbf{f}(\mathbf{x}, I) + b_i))}. \quad (24)$$

Intuitively, the algorithm tries to find the best shape, within the subspace of the shape model, such that the positions of the shape correspond to well-aligned (detected) parts (for more details the interested reader may refer to [24] and the references within for details regarding building and fitting CLM-based models). We have implemented CLMs using as features Histograms of Orientated Gradients (HoGs) and show that the proposed form of the one-shot similarity kernel in (9) with cutting planes SVMs is not only faster than CLMs in training but also increases the performance over SVMs with the one-shot similarity kernel (3) and standard SVMs. We should note here that the functional form of the one-shot similarity kernel (3) cannot be combined with the linear cutting-plane SVMs. For all one-shot similarity kernels the background samples for each of the points are selected as patches of other points (i.e., background samples for eyes were taken from the nose etc.).

We conducted experiments using the database that presents the challenge of uncontrolled natural settings. The Labeled Face Parts in the Wild (LFPW) database [3] consists of the URLs to 1100 training and 300 test images that can be downloaded from internet. All of these images were captured in the wild and contain large variations in pose, illumination, expression and occlusion. We were able to download only 813 training images and 224 test images as some URLs are no longer valid. These images were manually annotated with the 66-point markup to generate the ground-truths. In face deformable model fitting experiments, results are often reported in a curve of the proportion of the images vs the shape root mean square error (RMSE) between the predicted shape and the ground truth shape. We should note that the size of the faces in these images varies greatly due to the wild nature of this dataset. To overcome that, we normalized the shape RMSE by the distance between the eye-corners in order to show unbiased results. We compared with the state-of-the-art approach [37], which we trained using the same data and the code provided by the authors of [37]. Fig. 2 plots the curves for the all the tested methods. As we can see, the use of one-shot similarity kernels indeed increases the performance over standard SVMs. Furthermore, the use the closed form of one-shot similarity kernel (9) with the cutting-plane SVMs boosts even further the results over the original one-shot similarity kernel. Finally, in order to perform fair comparisons, we compared the actual time for solving the SVM optimization problem with the proposed kernel and either the cutting plane linear algorithm or the standard kernel SVM for the original version of the kernel. For the whole experiment, that is learning all 66 discriminant filters, one for each point. Training with the proposed kernel and the original form required 3 hours and around 1 day, respectively.

6. Conclusions

In this paper we studied a recently introduced class of kernels, the one-shot similarity kernels. We derived closed form feature maps and proved that they can be used for efficient exact incremental learning. We successfully combined them with typical classification algorithms (SVMs) and incremental learning techniques (iPCA) and applied them in several problems (face recognition, object tracking and deformable model fitting), acquiring state-of-the-art results. We verified their superiority not only in terms of computational complexity and time, but also of performance.

7. Acknowledgement

The work of Stefanos Zafeiriou and Irene Kotsia was partially funded by the EPSRC project EP/J017787/1 (4D-FAB). Irene Kotsia acknowledges support by the framework of the Action Supporting Postdoctoral Researchers of the Operational Program Education and Lifelong Learning (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

References

[1] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *IEEE T-PAMI*, 33(8):1619–1632, 2011.

[2] E. Bazavan, F. Li, and C. Sminchisescu. Fourier Kernel Learning. In *ECCV*, October 2012.

[3] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *CVPR*, pages 545–552. IEEE, 2011.

[4] T.-J. Chin and D. Suter. Incremental Kernel Principal Component Analysis. *IEEE T-IP*, pages 1662 – 1674, 2007.

[5] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE T-PAMI*, 25(5):564–577, 2003.

[6] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2:265–292, 2002.

[7] S. Günter, N. Schraudolph, and S. Vishwanathan. Fast Iterative Kernel Principal Component Analysis. *JMLR*, pages 1893 – 1918, 2007.

[8] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on Faces in'Real-Life'Images: Detection, Alignment, and Recognition*, 2008.

[9] T. Joachims. Training linear svms in linear time. In *ACM SIGKDD*, pages 217–226, 2006.

[10] T. Joachims, T. Finley, and C.-N. J. Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009.

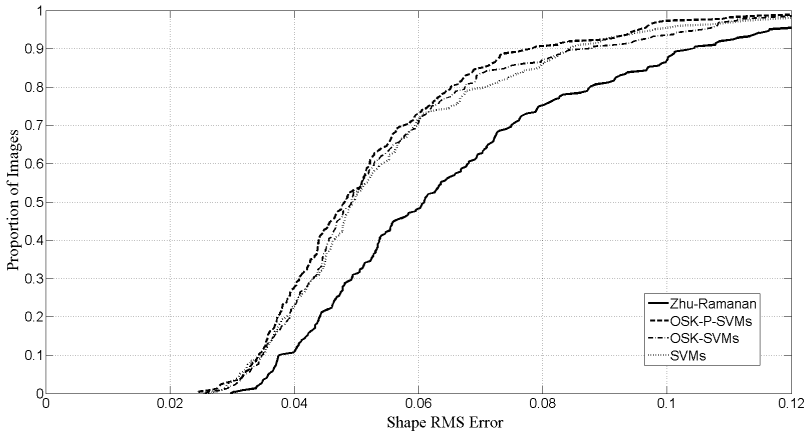

Figure 2. Shape RMS versus the proportion of the images

[11] K. I. Kim, M. O. Franz, and B. Scholkopf. Iterative kernel principal component analysis for image modeling. *IEEE T-PAMI*, 27(9):1351–1366, 2005.

[12] O. Kliper-Gross, T. Hassner, and L. Wolf. One shot similarity metric learning for action recognition. In *Proceedings of the First international conference on Similarity-based pattern recognition*, pages 31–45, 2011.

[13] I. Kotsia, I. Pitas, and S. Zafeiriou. Novel multiclass classifiers based on the minimization of the within-class variance. *IEEE Transactions on Neural Networks*, 20(1):14–34, 2009.

[14] A. Levy and M. Lindenbaum. Squential Karhunen-Loeve Basis Extraction and its Application to Images. *IEEE T-IP*, pages 1371 – 1374, 2000.

[15] F. Li, C. Ionescu, and C. Sminchisescu. Random fourier approximations for skewed multiplicative histogram kernels. *Pat. Rec.*, pages 262–271, 2010.

[16] F. Li, G. Lebanon, and C. Sminchisescu. Chebyshev Approximations to the Histogram χ^2 Kernel. In *CVPR*, 2012.

[17] S. Liwicki, S. Zafeiriou, G. Tzimiropoulos, and M. Pantic. Efficient online subspace learning with an indefinite kernel for visual tracking and recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 23(10):1624–1636, 2012.

[18] S. Maji, A. Berg, and J. Malik. Efficient classification for additive kernel svms. *IEEE T-PAMI*, pages 66–77, 2013.

[19] X. Mei and H. Ling. Robust visual tracking using l1 minimization. In *CVPR*, pages 1436–1443. IEEE, 2009.

[20] S. Nikitidis, S. Zafeiriou, and I. Pitas. Camera motion estimation using a novel online vector field model in particle filters. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8):1028–1039, 2008.

[21] F. Perronnin, J. Sánchez, and Y. Liu. Large-scale image categorization with explicit data embedding. In *CVPR*, pages 2297–2304. IEEE, 2010.

[22] A. Rahimi and B. Recht. Random features for large-scale kernel machines. *NIPS*, 20:1177–1184, 2007.

[23] D. Ross, J. Lim, R. Lin, and M. Yang. Incremental Learning for Robust Visual Tracking. *IJCV*, pages 125 – 141, 2008.

[24] J. M. Saragih, S. Lucey, and J. Cohn. Deformable model fitting by regularized landmark mean-shift. *IJCV*, 91(2):200–215, 2011.

[25] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[26] B. Schölkopf and A. J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[27] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE T-PAMI*, 34(3):480–492, 2012.

[28] C. Williams and M. Seeger. The effect of the input density distribution on kernel-based classifiers. In *ICML*, 2000.

[29] C. Williams and M. Seeger. Using the nystrom method to speed up kernel machines. *NIPS*, pages 682–688, 2001.

[30] L. Wolf, T. Hassner, and Y. Taigman. The one-shot similarity kernel. In *CVPR*, pages 897–902. IEEE, 2009.

[31] L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In *ACCV*, pages 88–97. 2010.

[32] L. Wolf, T. Hassner, and Y. Taigman. Effective unconstrained face recognition by combining multiple descriptors and learned background statistics. *IEEE T-PAMI*, 33(10):1978–1990, 2011.

[33] L. Wolf, T. Hassner, Y. Taigman, et al. Descriptor based methods in the wild. In *Workshop on Faces in'Real-Life'Images: Detection, Alignment, and Recognition*, 2008.

[34] S. Zafeiriou, A. Tefas, and I. Pitas. The discriminant elastic graph matching algorithm applied to frontal face verification. *Pattern recognition*, 40(10):2798–2810, 2007.

[35] S. Zafeiriou, A. Tefas, and I. Pitas. Learning discriminant person-specific facial models using expandable graphs. *IEEE T-IFS*, 2(1):55–68, 2007.

[36] S. Zafeiriou, A. Tefas, and I. Pitas. Minimum class variance support vector machines. *IEEE Transactions on Image Processing*, 16(10):2551–2564, 2007.

[37] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, pages 2879–2886, 2012.