A time series kernel for action recognition

Adrien Gaidon
http://lear.inrialpes.fr/people/gaidon
Zaid Harchaoui
http://lear.inrialpes.fr/people/harchaoui
Cordelia Schmid
http://lear.inrialpes.fr/people/schmid

LEAR - INRIA Grenoble, LJK
655, avenue de l'Europe
38330 Montbonnot, France

Abstract

We address the problem of action recognition by describing actions as time series of frames and introduce a new kernel to compare their dynamical aspects. Action recognition in realistic videos has been successfully addressed using kernel methods like SVMs. Most existing approaches average local features over video volumes and compare the resulting vectors using kernels on bags of features. In contrast, we model actions as time series of per-frame representations and propose a kernel specifically tailored for the purpose of action recognition. Our main contributions are the following: (i) we provide a new principled way to compare the dynamics and temporal structure of actions by computing the distance between their auto-correlations, (ii) we derive a practical formulation to compute this distance in any feature space deriving from a base kernel between frames and (iii) we report experimental results on recent action recognition datasets showing that it provides useful complementary information to the average distribution of frames, as used in state-of-the-art models based on bag-of-features.

1 Introduction

We address the problem of supervised action recognition, *i.e*. deciding whether an action is performed in a video, by learning video classifiers using non-linear Support Vector Machines (SVM). Such an approach allows to learn powerful classifiers by using only inner products in high-dimensional spaces, computed in practice *via* a kernel function. We propose here a new kernel specific to videos, which compares actions as time series of frames. Our kernel hinges upon the distance between auto-correlations to compare dynamic aspects of actions.

Following the progress on object recognition, significant improvements were observed on more and more challenging data coming from real-world video sources (*e.g*. sports [28] and Youtube videos [22]). Many approaches focus on extending successful ideas from related tasks on images (*e.g*. from object recognition [37]) and view videos as 3D spatio-temporal volumes. For instance, volumetric approaches include template-based techniques [12, 28, 32], tensor representations of videos [15] and local spatio-temporal features [19] accumulated in an orderless bag-of-features (BOF) model [7, 20, 30]. Though good results were achieved with models aggregating statistics of local features over the entire duration of an action (*c.f*. [33] for a recent evaluation), these action models were recently enhanced by treating

BMVC 2011 http://dx.doi.org/10.5244/C.25.63

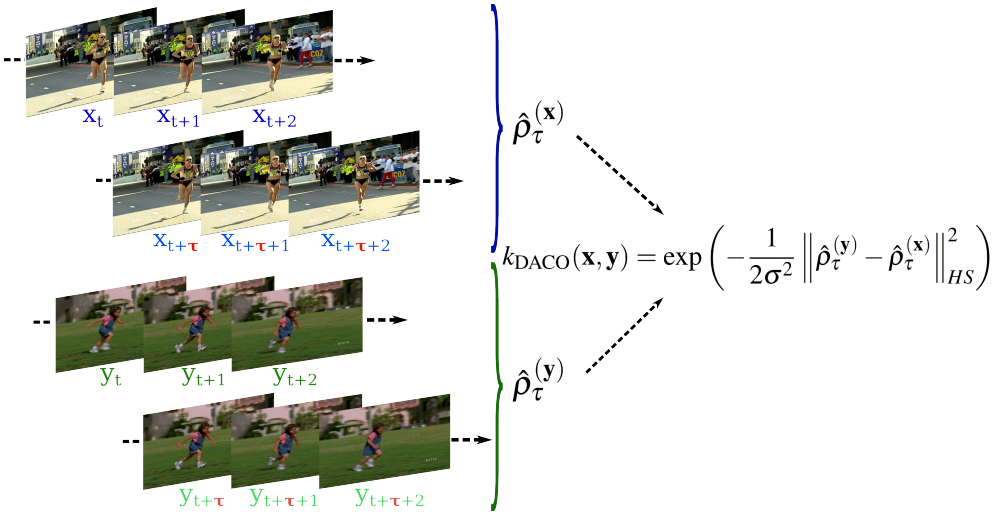

Figure 1: Computation of our DACO kernel. For two actions represented as time series of frames, $\mathbf{x} = (x_1, \ldots, x_T)$ and $\mathbf{y} = (y_1, \ldots, y_{T'})$, the kernel compares their dynamics by using the difference between their auto-correlations $\hat{\rho}_\tau^{(\mathbf{x})}$ and $\hat{\rho}_\tau^{(\mathbf{y})}$, with a lag of τ frames.

the time dimension differently from the spatial ones. For instance, Niebles *et al.* [25] propose a latent model of temporal parts for long duration activities and Gaidon *et al.* [8] model the temporal structure and ordering constraints of actions as sequences of "actoms". These methods only encode the coarse temporal structure and, therefore, fail to capture dynamic aspects like the temporal dependencies between frames. Another limitation of volumetric approaches is that the temporal granularity is not taken into account. Indeed, the temporal granularity among frames is much coarser than the spatial one (pixels), as fast discriminative motions can occur in a few frames.

The simplest way to represent actions and their temporal structure is just to concatenate, in the temporal order, per-frame feature vectors. Schindler and Van Gool [29] show that such a simple technique can yield good results in simple video conditions. However, their approach assumes that videos are synchronized in time beforehand. Consequently, it is not robust to significant variations in action speed. A more sophisticated approach based on chaos theory is used by Basharat and Shah [1]. They model repetitive human actions and dynamic textures as nonlinear dynamical systems. They use "strange attractors" to represent the dynamics of time series for action and dynamic texture synthesis, yet do not provide a way to compare two series of observations. Other approaches, inspired from speech and gesture recognition, represent actions as sequences of states [4, 13, 18, 26] or use dynamic probabilistic graphical models [2, 21, 35, 36] to model the temporal aspects of the videos. A limitation of these methods is that they only measure alignments between videos. Hence, they are not robust to partial observations (temporal occlusions) and significant duration variations. Furthermore, they generally involve a difficult intermediate recognition step, for instance by labeling each frame.

Both alignment-based approaches, computing a matching score between videos, and aggregation-based techniques, discarding most temporal aspects by averaging over frames, do not take into account characteristic dynamic information, like repeating patterns or the

relationships between frames. In contrast to these previous works, we represent actions directly as time series of frames and propose to model their key dynamic aspects by using auto-correlation, *i.e.* the cross-correlation of the signal of frames with a temporally shifted version of itself. Auto-correlation contains information pertaining to the temporal dependencies between frames and the temporal structure of actions, as it depends on the ordering of the frames. Hence, we propose to compare the dynamics of two actions by computing the distance between their respective auto-correlations. This distance is defined as the Hilbert-Schmidt norm of the difference between auto-correlations and we call the associated Gaussian RBF kernel the *Difference between Auto-Correlation Operators* (DACO) kernel (see figure 1 for an illustration). Note, that this is different from the cross-correlation between video volumes, which measures dependencies between frames of two different videos and is not suited to compare different actions with strongly related motions (*e.g.* running and walking).

DACO is also different from existing kernels on time series. Cuturi *et al.* [4, 5] proposed a kernel based on Dynamic Time Warping, with applications to speech recognition tasks. However, this kernel does not compare the dynamics but measures alignments between time series. Another example of time series kernel is given by Lu *et al.* [23], with applications to synchronized EEG segments. However, their similarity between two time series corresponds to similarity between temporal regularity, in the spirit of the functional data analysis framework [27].

We make the following contributions. We introduce our time series representation of videos and our novel DACO kernel in section 2. We give a practical formulation (section 2.3) that can operate on any type of frame model (including high-dimensional ones like BOF) by only requiring a kernel function on frames (*e.g.* the intersection kernel between histograms) and computing auto-correlation operators in the feature space induced by this kernel. Dynamic aspects alone are likely to be insufficient to describe some types of actions, especially when context or the nature of objects involved is discriminative. Therefore, our goal is to show that our DACO kernel is complementary with orderless aggregation statistics. Section 3 contains experimental results on recent action recognition datasets, showing that a simple linear combination of our DACO kernel and an aggregation-based kernel can improve recognition performance. Finally, some conclusions are given in section 4.

2 Auto-correlation kernel for time series of frames

Let $\mathscr{X}$ be a space of frame representations (*e.g.* histograms of visual words). Let a video $\mathbf{x}$ of duration T frames be represented as a time series $\mathbf{x} = (x_t)_{t=1\cdots T}$ where $x_t \in \mathscr{X}$. We define the space of videos $\mathscr{S} = \bigcup_{i>0} \mathscr{X}^i$. Our goal is to design a kernel $k_S : \mathscr{S} \times \mathscr{S} \to \mathbb{R}$ adapted to compare actions. In Section 2.1, we quickly recall the main properties of the auto-correlation, which is the base mathematical component of our approach. In Section 2.2, we give the definition and some details on our auto-correlation-based kernel. Finally, in Section 2.3, we derive a formulation of our kernel using only inner products between frames.

2.1 Auto-correlation

We compare the dynamic aspects of two actions $\mathbf{x} = (x_t)_{t=1\cdots T}$ and $\mathbf{y} = (y_t)_{t=1\cdots T'}$ by comparing their auto-correlations $\hat{\boldsymbol{\rho}}_\tau^{(\mathbf{x})}$ and $\hat{\boldsymbol{\rho}}_\tau^{(\mathbf{y})}$, *i.e.* their auto-covariances, $\hat{\boldsymbol{\Sigma}}_\tau^{(\mathbf{x})}$ and $\hat{\boldsymbol{\Sigma}}_\tau^{(\mathbf{y})}$, normalized

by their respective covariances, $\hat{\Sigma}^{(\mathbf{x})}$ and $\hat{\Sigma}^{(\mathbf{y})}$:

$$\hat{\rho}_\tau^{(\mathbf{x})} = \left(\hat{\Sigma}^{(\mathbf{x})} + \gamma \mathbf{I}\right)^{-1} \hat{\Sigma}_\tau^{(\mathbf{x})} \quad , \quad \hat{\rho}_\tau^{(\mathbf{y})} = \left(\hat{\Sigma}^{(\mathbf{y})} + \gamma \mathbf{I}\right)^{-1} \hat{\Sigma}_\tau^{(\mathbf{y})} \tag{1}$$

where γ is a regularization parameter and τ is the time lag in frames. The auto-covariance of a time series is simply defined as the cross-covariance between the series and a temporally shifted version of itself. For a time series (X_t) with mean $E[(X_t)] = \mu_t$, the auto-covariance at time t and lag τ is defined as

$$\Sigma^{(x)}(t, t+\tau) = E\left[(X_t - \mu_t)(X_{t+\tau} - \mu_{t+\tau})\right] = E\left[X_t X_{t+\tau}\right] - \mu_t \mu_{t+\tau} \tag{2}$$

Note that equation 1 makes the assumption that all time series are wide sense stationary, *i.e.* that first and second order moments do not vary with time. We show in our experiments (section 3) that this approximation yields reasonable results in practice for actions. We note the mean $\mu = \mu_t = \mu_{t+\tau}$ and the auto-covariance $\Sigma_\tau^{(x)} = \Sigma^{(x)}(t, t+\tau)$ which only depends on the lag τ. Their sampled versions estimated from the observation of the frames are noted $\hat{\mu}$ and $\hat{\Sigma}_\tau^{(\mathbf{x})}$. The auto-covariance contains information pertaining to temporal dependencies between frames, like repeating patterns. It is a special case of cross-covariance which has some interesting statistical properties [9]. Therefore, we propose to compare time series by computing the distance between their dynamics, modeled by the auto-correlation operators.

2.2 The DACO kernel

We define our *Difference between Auto-Correlation Operators* (DACO) kernel from the Hilbert-Schmidt norm of the difference between auto-correlations:

$$k_{\mathrm{DACO}}(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{1}{2\sigma^2} d_{\mathrm{DACO}}(\mathbf{x}, \mathbf{y})^2\right) \quad , \quad d_{\mathrm{DACO}}(\mathbf{x}, \mathbf{y}) = \left\| \hat{\rho}_\tau^{(\mathbf{y})} - \hat{\rho}_\tau^{(\mathbf{x})} \right\|_{HS} \tag{3}$$

The Hilbert-Schmidt norm, noted $\| \ \|_{HS}$, is simply the extension of the Frobenius matrix norm to any separable Hilbert space and can be defined for any bounded operator A as $\|A\|_{HS}^2 = \mathrm{Tr}(A^* A)$ where Tr denotes the trace function and A^* is the conjugate transpose of A. It derives from the Hilbert-Schmidt inner product $\langle A, B \rangle_{HS} = \mathrm{Tr}(A^* B)$. This allows us to decompose the DACO distance in three terms:

$$d_{\mathrm{DACO}}(\mathbf{x}, \mathbf{y})^2 = \left\| \hat{\rho}_\tau^{(\mathbf{x})} \right\|_{HS} + \left\| \hat{\rho}_\tau^{(\mathbf{y})} \right\|_{HS} - 2 \left\langle \hat{\rho}_\tau^{(\mathbf{y})}, \hat{\rho}_\tau^{(\mathbf{x})} \right\rangle_{HS} \tag{4}$$

As the norm of the auto-correlation operator measures the dependency between a series and a shifted version of itself, we can see from equation 4 that the distance will tend to be smaller for time series with almost no temporal structure (*e.g.* for random sequences of frames) and bigger for actions with quasi-deterministic relationships between neighboring frames (*e.g.* a constant translation movement). In addition, the inner product in equation 4 shows that if the dynamics of the two series $\mathbf{x}$ and $\mathbf{y}$ are different, then the distance will be bigger. Therefore, actions with strong but different temporal structures will tend to have large DACO distances. Consequently, DACO is well suited to compare actions characterized by their dynamic aspects, but needs to be combined with another kernel in order to deal with actions with little temporal structure. This shows that combining DACO with an aggregation-based kernel allows to efficiently represent dynamics and orderless distribution aspects, that are both useful for action recognition.

The time-lag parameter. The main parameter of our DACO kernel is the lag τ in frames. For periodic actions, it is important that this parameter be different from a multiple of the period. Indeed, a periodic signal is always perfectly correlated with its version shifted from a period and comparing two auto-correlations of two periodic signals with the same periodicity yields uninformative distances. This problem can be avoided in practice by multiple techniques, for instance by detecting the period or by averaging the distances for multiple τ values. However, these methods are generally either expensive or unreliable in the absence of prior information, *e.g.* for periodic actions with variable period durations. The alternative solution that we found to work best in practice is to simply take a τ which is small enough *w.r.t.* the action duration, such that it cannot be a multiple of a period. As we deal with short actions that include fast motions and potentially drastic changes in a few frames, we chose a τ of one frame in our experiments. This has the other advantage to not "dilute" temporal relations between the signal and its shifted version, hence ensuring the DACO distances are meaningful due to the preservation of strong temporal structures.

In the following section, we give a practical formulation of the DACO distance that is obtained by kernelizing equation 3, *i.e.* expressing it using only inner products between frames, computed via a kernel function.

2.3 Kernelized formulation of DACO

Frame representations are in general high-dimensional (*e.g.* several thousands of dimensions for BOF) and can be of a non-vector type (*e.g.* graphs). Therefore, instead of making assumptions on the frame models, we only assume the availability of a symmetric positive-definite kernel between frames $k_F : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$, such as the the intersection kernel between per-frame BOFs. We note $\mathscr{H}_F$ the feature space and $\phi_F : \mathscr{X} \to \mathscr{H}_F$ the feature map associated with the kernel $k_F(x_t, y_{t'}) = \langle \phi_F(x_t), \phi_F(y_{t'}) \rangle_{\mathscr{H}_F}$, between two frames x_t and $y_{t'}$ of two series $\mathbf{x}$ and $\mathbf{y}$. Note that $\mathscr{H}_F$ might be infinite-dimensional, for instance when the Gaussian RBF kernel is used. Yet, using the kernel trick, our kernel can be computed only by using kernel evaluations between frames.

In the following, we adopt notations similar to those of Shawe-Taylor and Cristianini [31]. Corresponding to the video $\mathbf{x}$, we define the time series $\mathbf{X}$ of frames in the frame feature space $\mathscr{H}_F$ with $\mathbf{X} = [\phi_F(x_1) \cdots \phi_F(x_T)]$ where the column t of $\mathbf{X}$ is the projection $\phi_F(x_t)$ of frame x_t. For two time series $\mathbf{x} = (x_t)_{t=1\cdots T}$, $\mathbf{y} = (y_{t'})_{t'=1\cdots T'}$ and their representations $\mathbf{X}$ and $\mathbf{Y}$, the matrix $\mathbf{K}$ of kernel evaluations between frames of both series is noted:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}^{(\mathbf{x})} & \mathbf{K}^{(\mathbf{x},\mathbf{y})} \\ \mathbf{K}^{(\mathbf{y},\mathbf{x})} & \mathbf{K}^{(\mathbf{y})} \end{bmatrix}, \quad \mathbf{K}^{(\mathbf{x})} = \mathbf{X}^T\mathbf{X}, \quad \mathbf{K}^{(\mathbf{y})} = \mathbf{Y}^T\mathbf{Y}, \quad \mathbf{K}^{(\mathbf{x},\mathbf{y})} = \mathbf{X}^T\mathbf{Y} = \left(\mathbf{K}^{(\mathbf{y},\mathbf{x})}\right)^T \quad (5)$$

Using our previous notations, we define the auto-covariance of action $\mathbf{x}$ at lag τ in the frame feature space $\mathscr{H}_F$ as:

$$\hat{\Sigma}_\tau^{(\mathbf{x})} = \frac{1}{T}\mathbf{X}\mathbf{X}_{+\tau}^T \quad \text{where } \mathbf{X}_{+\tau} = [\phi_F(x_{1+\tau}) \cdots \phi_F(x_{T+\tau})] \quad (6)$$

$\mathbf{x}^\tau = (x_{1+\tau}, \cdots x_{T+\tau})$ is the shifted version of $\mathbf{x}$ and $\mathbf{X}_{+\tau}$ is the corresponding time series representation in $\mathscr{H}_F$. Additionally, we define the kernel matrix $\mathbf{K}^{(\mathbf{x}^\tau)}$ between frames of $\mathbf{x}^\tau$. Note, that this formulation assumes that our actions have zero mean $\hat{\mu}_x = \frac{1}{T}\sum_{t=1}^T \phi_F(x_t)$. This requires centering the frames in the feature space $\mathscr{H}_F$. As our computations use only

kernel matrices, this is achieved by centering them directly [10, 31]:

$$\tilde{\mathbf{K}}^{(\mathbf{x})} = \tilde{\mathbf{X}}^T\tilde{\mathbf{X}} = \Pi_T \mathbf{K}^{(\mathbf{x})}\Pi_T, \quad \tilde{\mathbf{X}} = \mathbf{X}\Pi_T, \quad \Pi_T = \mathbf{I}_T - \frac{1}{T}\mathbf{1}_T\mathbf{1}_T^T \tag{7}$$

where $\mathbf{I}_T$ is the $T \times T$ identity matrix and $\mathbf{1}_T$ is the column vector of T ones. Π_T is called the centering matrix, $\tilde{\mathbf{X}}$ are the centered frames and $\tilde{\mathbf{K}}^{(\mathbf{x})}$ is the centered kernel matrix. In the rest of the paper, we always assume everything is centered in the feature space and use the notations $\mathbf{K}^{(\mathbf{x})}$ and $\mathbf{X}$ instead of $\tilde{\mathbf{K}}^{(\mathbf{x})}$ and $\tilde{\mathbf{X}}$.

We can compute the DACO kernel using only kernel matrices:

$$\begin{aligned} d_{\text{DACO}}(\mathbf{x},\mathbf{y})^2 &= \text{Tr}\left(\mathbf{N}^T\mathbf{K}\mathbf{N}\mathbf{K}_{+\tau}\right) \\ &= \text{Tr}\left(\mathbf{N}^{(\mathbf{x})^T}\mathbf{K}^{(\mathbf{x})}\mathbf{N}^{(\mathbf{x})}\mathbf{K}^{(\mathbf{x}^\tau)}\right) + \text{Tr}\left(\mathbf{N}^{(\mathbf{y})^T}\mathbf{K}^{(\mathbf{y})}\mathbf{N}^{(\mathbf{y})}\mathbf{K}^{(\mathbf{y}^\tau)}\right) \\ &\quad - 2\,\text{Tr}\left(\mathbf{N}^{(\mathbf{x})^T}\mathbf{K}^{(\mathbf{x},\mathbf{y})}\mathbf{N}^{(\mathbf{y})}\mathbf{K}^{(\mathbf{y}^\tau,\mathbf{x}^\tau)}\right) \end{aligned} \tag{8}$$

where $\mathbf{K}_{+\tau}$ is the $(T+T') \times (T+T')$ kernel matrix between all frames of the shifted series $\mathbf{x}^\tau$ and $\mathbf{y}^\tau$ and

$$\mathbf{N} = \begin{bmatrix} \mathbf{N}^{(\mathbf{x})} & \mathbf{0} \\ \mathbf{0} & -\mathbf{N}^{(\mathbf{y})} \end{bmatrix}, \quad \mathbf{N}^{(\mathbf{x})} = \frac{1}{\gamma^2 T}\left(\gamma\mathbf{I} - \left(T\mathbf{I} + \gamma^{-1}\mathbf{K}^{(\mathbf{x})}\right)^{-1}\mathbf{K}^{(\mathbf{x})}\right) \tag{9}$$

Proof. First, we recall that $\hat{\Sigma}^{(\mathbf{x})} = \frac{1}{T}\mathbf{X}\mathbf{X}^T$ and $\mathbf{K}^{(\mathbf{x})} = \mathbf{X}^T\mathbf{X}$. We use the Sherman-Morrison-Woodbury formula to develop the normalization factors:

$$\begin{aligned} \left(\hat{\Sigma}^{(\mathbf{x})} + \gamma\mathbf{I}\right)^{-1} &= \left(\gamma\mathbf{I} + \mathbf{X}\frac{1}{T}\mathbf{I}\mathbf{X}^T\right)^{-1} \\ &= \frac{1}{\gamma^2}\left(\gamma\mathbf{I} - \mathbf{X}\mathbf{Q}^{(\mathbf{x})}\mathbf{X}^T\right) \quad \text{where } \mathbf{Q}^{(\mathbf{x})} = \left(T\mathbf{I} + \frac{1}{\gamma}\mathbf{K}^{(\mathbf{x})}\right)^{-1} \end{aligned} \tag{10}$$

Using the fact that $\hat{\Sigma}^{(\mathbf{x})}_\tau = \frac{1}{T}\mathbf{X}\mathbf{X}^T_{+\tau}$, equation 10 allows us to re-write the auto-correlation as:

$$\left(\hat{\Sigma}^{(\mathbf{x})} + \gamma\mathbf{I}\right)^{-1}\hat{\Sigma}^{(\mathbf{x})}_\tau = \mathbf{X}\mathbf{N}^{(\mathbf{x})}\mathbf{X}^T_{+\tau} \quad \text{where } \mathbf{N}^{(\mathbf{x})} = \frac{1}{\gamma^2 T}\left(\gamma\mathbf{I} - \mathbf{Q}^{(\mathbf{x})}\mathbf{K}^{(\mathbf{x})}\right) \tag{11}$$

We then compute the Hilbert-Schmidt norm of the difference between auto-correlations by expanding it from the Hilbert-Schmidt inner product (*c.f.* equation 4):

$$d_{\text{DACO}}(\mathbf{x},\mathbf{y})^2 = \left\|\mathbf{X}\mathbf{N}^{(\mathbf{x})}\mathbf{X}^T_{+\tau}\right\|^2_{HS} + \left\|\mathbf{Y}\mathbf{N}^{(\mathbf{y})}\mathbf{Y}^T_{+\tau}\right\|^2_{HS} - 2\left\langle\mathbf{X}\mathbf{N}^{(\mathbf{x})}\mathbf{X}^T_{+\tau}, \mathbf{Y}\mathbf{N}^{(\mathbf{y})}\mathbf{Y}^T_{+\tau}\right\rangle_{HS} \tag{12}$$

The Hilbert-Schmidt norm of the auto-correlation is computed from the trace definition by:

$$\begin{aligned} \left\|\mathbf{X}\mathbf{N}^{(\mathbf{x})}\mathbf{X}^T_{+\tau}\right\|^2_{HS} &= \text{Tr}\left(\left(\mathbf{X}\mathbf{N}^{(\mathbf{x})}\mathbf{X}^T_{+\tau}\right)^T\mathbf{X}\mathbf{N}^{(\mathbf{x})}\mathbf{X}^T_{+\tau}\right) \\ &= \text{Tr}\left(\mathbf{N}^{(\mathbf{x})^T}\mathbf{X}^T\mathbf{X}\mathbf{N}^{(\mathbf{x})}\mathbf{X}^T_{+\tau}\mathbf{X}_{+\tau}\right) \quad (13) \\ &= \text{Tr}\left(\mathbf{N}^{(\mathbf{x})^T}\mathbf{K}^{(\mathbf{x})}\mathbf{N}^{(\mathbf{x})}\mathbf{K}^{(\mathbf{x}^\tau)}\right) \quad (14) \end{aligned}$$

where equation 13 results from the fact that the trace of products is invariant to circular permutations. The Hilbert-Schmidt inner product in equation 12 is obtained using the same approach. This completes the proof of equation 8. □

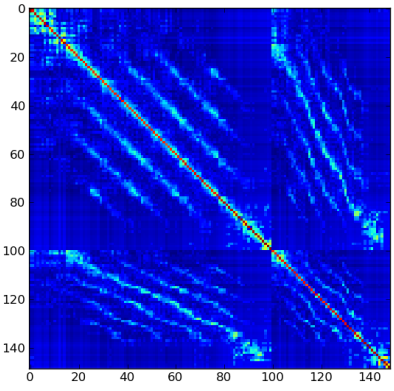

Figure 2: Example of a kernel matrix $\mathbf{K}$ (*c.f.* equation 5) of kernel evaluations between the frames of two "walking" actions from the KTH dataset. "Warmer" colors represent higher similarity, the axes represent time (in frames) and the four blocks correspond to (from top-left to bottom-right) $\mathbf{K}^{(\mathbf{x})}$, $\mathbf{K}^{(\mathbf{x},\mathbf{y})}$, $\mathbf{K}^{(\mathbf{y},\mathbf{x})}$ and $\mathbf{K}^{(\mathbf{y})}$. Note that actions are both periodic and display similar structures (visible in off-diagonal blocks).

Our kernel is comparing time series using only between-frame kernel matrices as described in equation 8. Therefore, it depends on the number of frames (typically of the order of 100 frames per action), instead of the number of dimensions of frame descriptors (of the order of 10 000 for BOF). Furthermore, Junejo *et al.* [12] observed that $\mathbf{K}$ is a matrix of "temporal self-similarities" and has some interesting properties for action recognition, namely its stability with respect to view point changes and its action specific structure (*c.f.* figure 2 for an example of such a kernel matrix). However, Junejo *et al.* [12] propose to represent the structure of this matrix by viewing it as an image described using HOG features. In a way, we show that these temporal self-similarities are related to auto-correlation operators in the feature space associated with the frame kernel.

3 Experiments

In this section, we evaluate our approach using non-linear SVMs on standard video benchmarks for action classification. We first introduce the datasets we use, then describe how we model frames and the base kernel between these representations. We then describe how we combine a simple aggregation-based kernel with our DACO kernel. Finally, we give the classification results of our approach and compare it to related and state of the art methods.

3.1 Datasets

As dynamic aspects of actions might not be useful to classify every type of action, we investigate the use of our kernel on three state of the art datasets.

The **KTH** dataset [1] [30] is composed of six human action categories: three similar displacement ones (walking, jogging and running) and three others involving mostly arm motions (boxing, waving and hand-clapping). Note that these actions are periodic. This dataset contains 2391 videos filmed with four different scenarios but with homogeneous and static backgrounds (in most sequences). We use the evaluation protocol of [30]: accuracy averaged over all classes, for a fixed train and test split.

[1] http://www.nada.kth.se/cvap/actions/

The **UCF Sports** dataset [2] [28] contains ten human actions: swinging (on the pommel horse and on the floor) diving, kicking (a ball), weight-lifting, horse-riding, running, skateboarding, swinging (at the high bar), golf swinging and walking. The dataset consists of 150 video samples which show a large intra-class variability. To increase the amount of data samples, the dataset is extended with horizontally flipped versions of each sequence. Videos of this dataset are of high resolution and good quality. The evaluation metric is the average leave-one-out accurary (without considering the flipped versions at test time).

The **Youtube** dataset [3] [22] contains eleven action categories: basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. This dataset is challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background and illumination conditions. Videos are of low resolution, contain shaky camera motion and actions are characterized not only by motion but also by the objects involved and the context in which they are performed. Performance is measured as in [22] using "leave-one-group-out" average accuracy.

For all experiments, the predictions for the different kernels are obtained by learning non-linear SVMs with the standard "one-against-rest" multi-class approach.

3.2 Frame description and kernel

We used the features recently proposed by Wang *et al.* [34], which achieved the best performance so far, with a simple BOF, on the three datasets mentioned above. First, we compute densely sampled local feature trajectories from the dense optical flow field between frames. Then, for each feature track, a concatenation of trajectory-aligned local descriptors is computed. We use the best descriptor reported in [34]: Motion Boundary Histograms (MBH [6]), which is the quantized spatial derivatives of the horizontal and vertical components of the optical flow. We use the same track and descriptor parameters as the ones mentioned in [34], namely feature tracks of 15 frames, with a dense sampling stride of 5 pixels. We then represent each frame by a BOF. We compute a dictionary of 4000 "visual words" obtained with k-means clustering on a subset of 100,000 randomly sampled features (separately for each dataset). Each feature is assigned to a visual word. Then, each frame is modeled with the histogram of occurrences of visual words corresponding to tracks passing through this frame. Note that our frame representations depend on neighboring ones and duplicating visual words along tracks is a way to perform temporal smoothing. We obtain a video representation as a time series of per-frame BOF. In the rare cases where the histogram of a frame is empty, we replace it by its linear interpolation obtained from neighboring frames. We use the intersection kernel between histograms [24] as base kernel between frames.

3.3 Combination with an aggregation-based kernel

In order to show the complementarity of our DACO kernel with traditionally used aggregation-based kernels, we provide results for a kernel that is the linear combination of our DACO kernel and the *Difference between Mean Elements* (DME) kernel:

$$k_{DME}(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{1}{2\sigma^2} d_{DME}(\mathbf{x},\mathbf{y})^2\right) \quad , \quad d_{DME}(\mathbf{x},\mathbf{y}) = \left\|\hat{\mu}_y - \hat{\mu}_x\right\|_{\mathscr{H}_F} \tag{15}$$

This kernel is simply using the difference between the means, $\hat{\mu}_x$ and $\hat{\mu}_y$, of the frames in the feature space $\mathscr{H}_F$. This is related to the traditional BOF approach consisting of aggregating

[2] http://www.cs.ucf.edu/vision/public_html/

[3] http://www.cs.ucf.edu/ liujg/YouTube_Action_dataset.html

KTH		UCF Sports		Youtube	
Niebles *et al.* [25]	91.3	Wang *et al.* [33]	85.6	Liu *et al.* [22]	71.2
Brendel *et al.* [3]	94.2	Kläser *et al.* [16]	86.7	Ikizler *et al.* [11]	75.21
Kovashka *et al.* [17]	94.53	Kovashka *et al.* [17]	87.27	Brendel *et al.* [3]	77.8
Wang *et al.* [34]	**95.0**	Wang *et al.* [34]	84.8	Wang *et al.* [34]	83.9
DME	94.8	DME	87.0	DME	86.7
DACO	93.4	DACO	85.3	DACO	79.1
DME + DACO	**94.9**	DME + DACO	**90.3**	DME + DACO	**87.9**

Table 1: Average accuracy (in %) on KTH [30], UCF Sports [28] and Youtube [22].

local descriptors computed over the entire video sequence like in [20, 34], except that the aggregation is performed in the feature space. In practice, the DME kernel is computed using the same kernel matrix $\mathbf{K}$ (before centering) as DACO by:

$$d_{DME}(\mathbf{x},\mathbf{y})^2 = \mathbf{m}^T\mathbf{K}\mathbf{m} \quad , \quad \mathbf{m} = \Big[\underbrace{-\frac{1}{T},\cdots,-\frac{1}{T}}_{T},\underbrace{\frac{1}{T'},\cdots,\frac{1}{T'}}_{T'}\Big]^T \tag{16}$$

In the following section, we report results for the DME kernel, the DACO kernel and the linear combination of the two (referenced as DME+DACO below).

3.4 Results

In table 1, we report average classification accuracies for state-of-the-art methods on the three datasets mentioned in section 3.1. Note that, for a fair comparison, we report the results of Wang *et al.* [34] obtained with MBH features only. We achieve state-of-the-art performance on all datasets with the simple combination of the aggregation-based DME kernel and our auto-correlation-based DACO kernel. Furthermore, the combination DME+DACO is similar (on KTH) or superior to the best one of the two. Using the same visual features, we improve over the results of Wang *et al.* [34] by +5.5% on UCF Sports (+2.1% *w.r.t.* state of the art [34]) and by +4% on Youtube (+3.7% *w.r.t.* state of the art [34]). These experimental results suggest that the two kernels complement each other for the purpose of action recognition.

The results of DACO on the Youtube dataset could be explained by the longer duration of actions in this dataset (approximatively 160 frames on average) compared to the duration of actions in UCF Sports (around 60 frames on average). This is confirmed by the good performance of DACO on KTH and UCF Sports. This shows that DACO is more suited to short duration, fast actions and is explained by: (i) the small value of the lag we use ($\tau = 1$ frame) and (ii) long range temporal dependencies between frames of non-periodic actions are difficult to estimate. This suggests a possible improvement by applying our DACO kernel in a more temporally localized manner, in order to detect correlated components with a strong temporal structure.

4 Conclusions

This paper has introduced a new kernel on videos, the *Difference between Auto-Correlation Operators* (DACO) kernel, specifically designed for action recognition. It compares the dynamics of actions, represented as time series of frames, by using a distance between auto-correlations. It can be efficiently computed using only inner products between frames.

We show that, even if not all actions exhibit characteristic temporal relationships between frames, the dynamic information extracted by auto-correlations can complement state-of-the-art distribution-based kernels that average visual information over frames.

As suggested by our experiments, DACO is more suited to short actions characterized by their dynamics. Future work will investigate other video sources *a priori* more adapted to our time series approach, for instance, video-surveillance scenarii with low frame-rates.

Acknowledgments This work was partially funded by the MSR/INRIA joint project, the European integrated project AXES and the PASCAL 2 Network of Excellence.

References

[1] A. Basharat and M. Shah. Time series prediction by chaotic modeling of nonlinear dynamical systems. In *CVPR*, 2009.

[2] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *CVPR*, 1997.

[3] W. Brendel and S. Todorovic. Activities as time series of human postures. In *ECCV*, 2010.

[4] M. Cuturi. Fast global alignment kernels. ICML, 2011.

[5] M. Cuturi, J.-P. Vert, O. Birkenes, and T. Matsui. A kernel for time series based on global alignments. In *ICASSP*, 2007.

[6] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006.

[7] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, pages 65–72, 2005.

[8] A. Gaidon, Z. Harchaoui, and C. Schmid. Actom sequence models for efficient action detection. In *CVPR*, 2011.

[9] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *Algorithmic Learning Theory*, pages 63–77, 2005.

[10] Z. Harchaoui, F. Bach, and E. Moulines. Testing for homogeneity with kernel Fisher discriminant analysis. In *Adv. NIPS*, 2008.

[11] N. Ikizler-Cinbis and S. Sclaroff. Object, scene and actions: combining multiple features for human action recognition. In *ECCV*, 2010.

[12] I. N. Junejo, E. Dexter, I. Laptev, and P. Perez. View-independent action recognition from temporal self-similarities. *PAMI*, 2010.

[13] S.H. Jung, Y. Guo, H. Sawhney, and R. Kumar. Action video retrieval based on atomic action vocabulary. In *MIR*, 2008.

[14] Y. Ke, R. Sukthankar, and M. Hebert. Spatio-temporal shape and flow correlation for action recognition. In *CVPR*, 2007.

[15] T.K. Kim and R. Cipolla. Canonical correlation analysis of video volume tensors for action categorization and detection. *PAMI*, pages 1415–1428, 2008.

[16] A. Kläser, M. Marszalek, I. Laptev, and C. Schmid. Will person detection help bag-of-features action recognition? Research report, INRIA, 2010.

[17] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *CVPR*, 2010.

[18] K. Kulkarni, E. Boyer, R. Horaud, and A. Kale. An unsupervised framework for action recognition using actemes. In *ACCV*, 2010.

[19] I. Laptev. On space-time interest points. *IJCV*, 64(2–3):107–123, 2005.

[20] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[21] B. Laxton, J. Lim, and D. Kriegman. Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video. In *CVPR*, 2007.

[22] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *CVPR*, 2009.

[23] Z. Lu, T.K. Leen, Y. Huang, and D. Erdogmus. A reproducing kernel Hilbert space framework for pairwise time series distances. In *ICML*, 2008.

[24] S. Maji, A. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2008.

[25] J.C. Niebles, C. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010.

[26] S. Nowozin, G. Bakir, and K. Tsuda. Discriminative Subsequence Mining for Action Classification. In *ICCV*, 2007.

[27] J.O. Ramsay and B.W. Silverman. *Functional Data Analysis*. Springer, 2005.

[28] M.D. Rodriguez, J. Ahmed, and M. Shah. Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

[29] K. Schindler and L. Van Gool. Action snippets: How many frames does human action recognition require. In *CVPR*, 2008.

[30] C. Schueldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *ICPR*, 2004.

[31] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge Univ Pr, 2004.

[32] E. Shechtman and M. Irani. Space-time behavior based correlation. In *CVPR*, 2005.

[33] H. Wang, M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[34] H. Wang, A. Kläser, C. Schmid, and L. Cheng-Lin. Action recognition by dense trajectories. In *CVPR*, 2011.

[35] J. Yamato, J. Ohaya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *CVPR*, 1992.

[36] Z. Zeng and Q. Ji. Knowledge based activity recognition with dynamic bayesian network. In *ECCV*, 2010.

[37] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 73(2):213–238, 2007.