A Time Flexible Kernel Framework for Video-Based Activity Recognition

Mario Rodriguez[a,*], Carlos Orrite[a], Carlos Medrano[b,a], Dimitrios Makris[c]

[a] *CVLab, I3A, Zaragoza University, c/Mariano Esquillor s/n 50018, Zaragoza, Spain*
[b] *EduQTech, E.U. Politecnica, Zaragoza University, c/Ciudad Escolar s/n, 44003, Teruel, Spain*
[c] *Digital Imaging Research Centre, Kingston University, Penrhyn Road Kingston upon Thames Surrey KT1 2EE,UK*

Abstract

This work deals with the challenging task of activity recognition in unconstrained videos. Standard methods are based on video encoding of low-level features using Fisher Vectors or Bag of Features. However, these approaches model every sequence into a single vector with fixed dimensionality that lacks any long-term temporal information, which may be important for recognition, especially of complex activities. This work proposes a novel framework with two main technical novelties: First, a video encoding method that maintains the temporal structure of sequences and second a Time Flexible Kernel that allows comparison of sequences of different lengths and random alignment. Results on challenging benchmarks and comparison to previous work demonstrate the applicability and value of our framework.

Keywords: Activity Recognition, Soft-assignment, Kernel Methods, Support Vector Machine

[*] Corresponding author
Email addresses: mrodrigo@unizar.es (Mario Rodriguez), corrite@unizar.es (Carlos Orrite), ctmedra@unizar.es (Carlos Medrano), d.makris@kingston.ac.uk (Dimitrios Makris)

Abbreviations:

Time Flexible Kernel (TFK), Improved Dense Trajectories (IDT), Motion Interchange Patterns (MIP), Probability Product Kernel (PPK)

1. Introduction

Significant research effort has been invested in video-based activity recognition during the last few years, supported by the widespread availability of video cameras, as it may benefit many applications such as video indexing, surveillance or entertainment.

A video is a sequence of frames that can be viewed as a 3-dimensional matrix of pixels, two dimensions provide the space localization and the third one is related to time. When displayed in a screen, as a sequence of images, humans are able to easily distinguish among activities, but the same task is extremely challenging for a computational method. The machine learning community has suggested several approaches with their advantages and disadvantages, but results in unconstrained recordings are still far from what humans may achieve.

The design of sophisticated low-level descriptors [1] [2] [3] has been central in recent advances for this research challenge. Specifically, space-time feature codification has been present in the state-of-the-art approaches where video sequences are represented by a Bag of Features (BoF) or a Fisher Vector (FV), encoding the extracted features. The recognition process is carried out afterwards by applying a multi-class Support Vector Machine (SVM) [4], which takes advantage of the kernel trick. Despite the promising performance of these approaches there are two drawbacks due to the characteristics of the descriptors: (i) using image or short-term descriptors, the lack of explicit temporal information withhold them from reliable recognition of activities [5], and (ii) mid-term descriptors may describe better the activities [6] but still lack of information of the whole temporal structure making them unreliable for complex activities where the order of sub-actions describes the activity.

On the other hand, some state space models such as Hidden Markov Models (HMM) [7] or more recent Conditional Random Fields [8], codify the long term temporal information of the sequences. Although they have provided satisfactory results in human activities they usually work in constrained scenarios such as ones implied by the datasets KTH, Weizmann or UT-Tower, where there

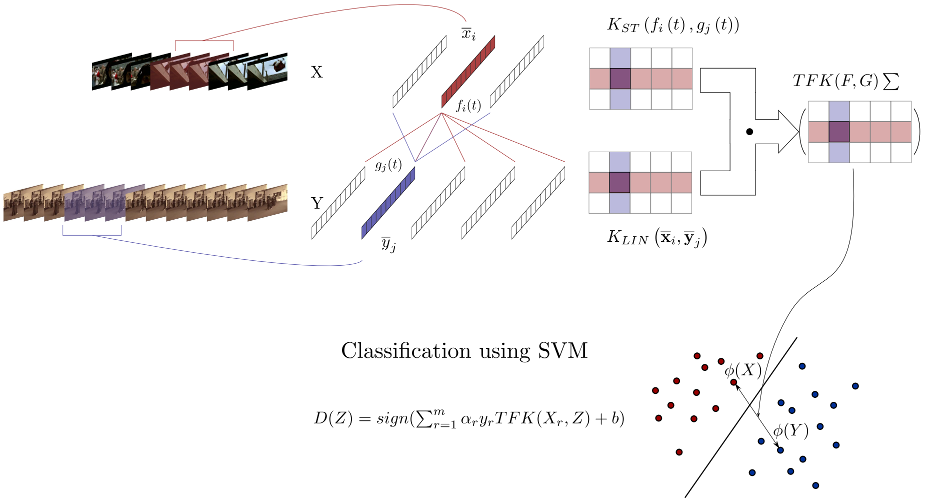

$\overline{x}_i$
X
$f_i(t)$
$g_j(t)$
Y
$\overline{y}_j$
$K_{ST}\left(f_i\left(t\right), g_j\left(t\right)\right)$
$K_{LIN}\left(\overline{\mathbf{x}}_i, \overline{\mathbf{y}}_j\right)$
$TFK(F,G)\sum$
Classification using SVM
$D(Z) = sign(\sum_{r=1}^{m} \alpha_r y_r TFK(X_r, Z) + b)$
$\phi(X)$
$\phi(Y)$

Figure 1: Graphical Abstract: (i) sliding frame-windows in the clip frames are used to encode the low-level descriptors information obtaining a sequence of vectors, (ii) two different kernels between sequence elements are computed using video and structure information and (iii) a multiple kernel learning is calculated in order to recognize using a SVM.

is no camera motion and the point of view is fixed [9] [10] [11]. Thanks to these restrictions it is possible to train states encompassing common characteristics among the videos and thus to achieve high accuracy. However, new databases, such as HMDB51, UCF50, OlympicSports and Virat Release 2.0 have been produced aiming at challenging tasks, such as indexing events in unconstrained videos in the internet or surveillance in uncontrolled scenarios performing a scene-independent learning and recognition. These datasets were recorded in unconstrained environments with random viewpoints, camera movements and/or dynamic changes in the background.

Some of the best results in these challenging benchmark datasets have been obtained with variations of the mentioned SVM approach [12] [4] [6], which has been proven to be a convenient method in spite of the lack of long-term dynamic information. Nevertheless, the long-term temporal information is important in the description of complex activities and thus, we propose the recognition framework depicted in Figure 1 where such information is maintained. Both BoF and

FV create a codebook, the former using k-means clustering of the training descriptors and the latter using an EM-GMM algorithm. Following the application of sliding frame-windows, each video is modelled as a sequence of BoFs or FVs. The use of a window with few frames produces sparse data so we minimize its effect by using a soft-assignment approach when using BoFs. These sequences preserve the long-term dynamic information needed for the recognition of complex activities. However, as the sequences length and the pace of actions are variable, standard kernels obtained between vectors of same length are not applicable and novel approaches, like Spatio-Temporal Pyramid Matching (STPM) [13], keep the long-term information, but they rely on perfect alignment of the sequences with regular pace of actions. Nevertheless, it is worth noting the improvement achieved using several encoding scales, proposed in STPM, that we also apply into our work. So, our contribution in this paper includes the design of a novel kernel formulation between arbitrary length sequences that allows the use of the long-term dynamic information in a SVM with matching flexibility, named Time Flexible Kernel (TFK). In order to validate our contribution we have carried out several experiments in four challenging datasets: HMDB51, UCF50, OlympicSports and Virat Release 2.0.

The rest of the paper is divided as follows. Section 2 reviews some related works. Section 3 explains the proposed framework, focusing on our two main technical contributions: a novel encoding scheme and the Time Flexible Kernel, as well as its application for activity recognition. Section 4 presents our experimental validation and section 5 concludes the work.

2. Related Work

Feature extraction is a key element in recognition systems thus a significant number of methods have been proposed. Descriptors can be divided into global and local or low-level features, however, the former group is sensitive to noise, occlusions and viewpoint variations that can be experienced in challenging datasets such as HMDB51, UCF50, OlympicSports and Virat Release

2.0. Therefore, researchers have mainly focused on local descriptors that can be roughly categorised as: (i) image descriptors like SIFT, HOG, Harris, which lack any kind of time information or (ii) space-time descriptors like their extensions 3D-SIFT [2], HOG3D [3] and spatio-temporal Harris interest points [1]. Some descriptors of the latter group are designed so to capture directly the video information as Motion Interchange Patterns (MIP) [5], HOG-HOF [14] or SCISA [15] do. Most of the approaches use holistic encoding based on BoF or FV to derive a vector for each video sequence. Then, SVM is used for multi-class classification, either using a one-against-one approach or a one-against-all approach for multi-class recognition. Until recently, BoF has been the standard approach in video encoding, quantizing the feature vectors into a predefined codebook. However, recent works [6] [4] have adopted the strategy of encoding the feature information into a FV which keeps second-order information in relation to an estimated Gaussian Mixture Model (GMM). Alternatively to the above framework, [16] modifies the SVM strategy to a ranking problem obtaining encouraging results. Finally, Dense Trajectories [17] [18] and the actual state of the art in many benchmarks Improved Dense Trajectories (IDT)[6] are based on short trajectories of local descriptors within a sliding temporal window.

The activity encoding methods such as BoF and FV require a “visual word” dictionary. Usually, such a dictionary is created using clustering of the extracted features in the training examples. In the case of BoF every feature is roughly assigned to a word of the dictionary. However, quantization errors are caused, when only a small amount of samples are used, as in the case of the use of a narrow temporal window. Soft-assignment approaches have been proposed to deal with this problem [19] [20] [21]. Alternatively, FV models the codebook as a GMM and the encoding produces a vector of $K(2d + 1)$ dimensions, K is the number of Gaussians and d is the dimension of descriptors, where second order information of the GMM is used [4] [6]. However, the main drawback of both approaches is that the long-term temporal information is lost since the encodings are obtained from an unordered collection of local descriptors.

Many approaches have been designed to account complex temporal struc-

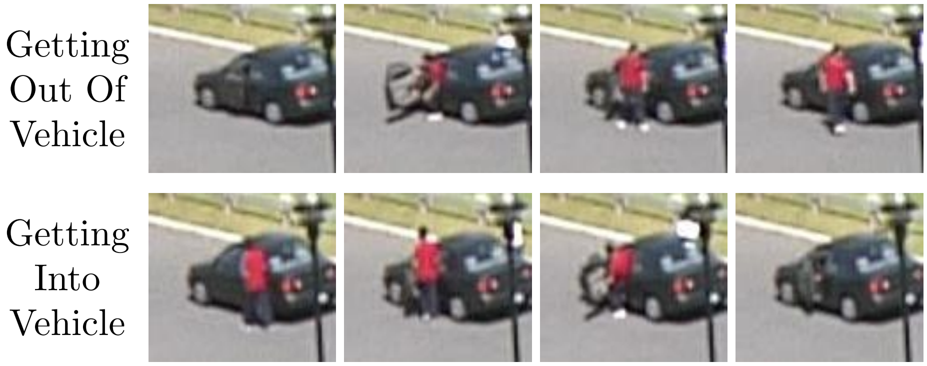


Figure 2: Two "opposite" activities from Virat Release 2.0 dataset with different sub-actions order. First row: person opens car door, goes out of the vehicle, closes door and walks. Second row: person walks, opens car door, gets into the vehicle and closes door.

tures recognizing complex human activities defined as composite multimedia semantics (e.g., birthday party, wedding ceremony) where orderless sub-actions appear in the video. These approaches consider the order of sub-actions as a distracter (not a discriminant) and hence they perform an alignment of similar sub-scenes disregarding their order. For instance *Xu et al.* [22] divide the clips into sub-clips which later are matched with other sequence using the earth movers distance (EMD). *Cao et al.* [23] have designed a kernel that makes a pooling of the frames into a fixed number of scenes called Scene Alignment Pooling (SAP). In [24] a detection of sub-scenes categories and a global scene category are combined in a Multiple Kernel Latent SVM where several features are used. In the work of *Li et al.* [25] the proposed method identifies the most representative segments of the actions using a dynamic pooling with a latent variable.

As opposed to the previously explained works, the temporal order of the sub-actions performed in an activity is considered essential in this paper, as the objective is to distinguish between complex activities that can be composed of same sub-actions but in different order, even opposite as for instance "Getting Out of Vehicle" and "Getting Into Vehicle" in Virat dataset, Figure 2.

Thanks to the kernel trick the SVM can classify in a dimensional space different from the original one where samples may be linear-separable. The standard kernel methods assume a fixed length D-dimensional vector per sample which is projected into a different space where the inner product is performed. However, this is not straightforward in activity recognition videos where the long-term activity dynamic information remains in the encoding because lengths of sequences may be arbitrary. Two solutions have been proposed in the literature: (i) obtaining some sort of inner product by aligning the sequences lengths of the patterns, as Dynamic Time-Alignment Kernel [26] or Fast Global Alignment Kernel [27] do and (ii) training a HMM with a single sequence and posterior obtaining a Probability Product Kernel (PPK) [28] like in [29]. Both solutions have been used in sequence clustering tasks [30] [29] [31]. Sequence alignment enforces a common start and end in the sequences which is not always the case. On the other hand, the PPK of HMMs implies that each HMM is trained with only one sequence which does not offer sufficient information to train properly the parameters of a complex model. Moreover, the optimization process in the HMM training is performed with the Baun-Welch algorithm which only assures a local optimum.

The long-term temporal information has been used in some other methods. A recent one is the extension of the work of Spatial Pyramid Matching (SPM) [32] called Spatio-Temporal Pyramid Matching (STPM) [13]. The method suggests dividing the videos into equal number of spatio-temporal volumes at several scales, called pyramids, computing in each volume a BoF, and finally obtaining a similarity between two video clips by comparing the corresponding volumes. Fixing the number of divisions and comparing volumes one-to-one constrain the method to regular paced actions losing flexibility. In [33] they encode each video into a 3D histogram with spatio-temporal information and design a specific kernel for the 3D histogram, but their pairwise feature comparison is not suitable for dense features extraction which recently have provide the state-of-the-art results. Using HMM, the authors of [34] propose to learn the temporal structure, including latent variables that determine the expected time to stay in a state. [35] keeps the long-term information using graph models of the video foreground and with a Kronecker product constructs a Kronecker graph model per action class used in the classification. In [36] the sequences length is not constrained and the authors use a framework where appearance and temporal position of motion segments is included, however, due to high computational load, their learning process finds a local optimum.

As pace of actions is unknown and the alignment performed by the segmentation can introduce some errors, our approach improves the previous methods by introducing some flexibility in the kernel framework that compensates in some degree these errors. The framework allows arbitrary length vectors in the kernel, and as the SVM training is a convex optimization problem it finds a global optimum.

3. Activity Recognition Framework using a Time Flexible Kernel (TFK)

Figure 3 summarises our proposed framework for activity recognition and compares it to the standard approach. Specifically, it assumes a pipeline of feature extraction and clustering, video encoding using BoF or FV, applying kernel and finally using multi-class SVM. The contribution of our framework is twofold: First, the video encoding method that produces multiple encodings that preserve temporal information, as described in section 3.1. Second, the Time Flexible Kernel that allows comparison of vectors of different lengths, as described in section 3.2. Section 3.3 discusses the application of the novel framework on activity recognition.

3.1. Video Encoding

We have designed an encoding that maintains the temporal information of sequences. In Figure 3 we can compare the standard approach (up) and our encoding method (down). There is a common stage of features extraction and codebook generation by clustering, but the video is represented differently. Our proposal keeps temporal information by computing the FV or BoF on sliding

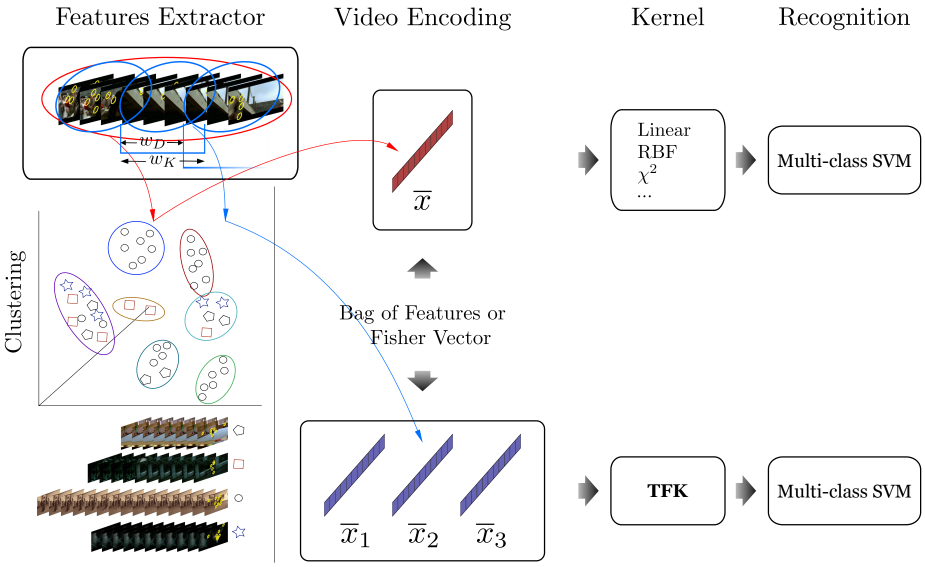


Figure 3: Standard (up) and proposed (down) approaches: The features extraction and the clustering stages are common. The standard approach encodes a video into a single BoF using hard-assignment to clusters or a single FV. Our novel approach encodes the video splitting it into sliding frame-windows (window duration, w_K frames, and window stride, w_D frames) obtaining a BoF using soft-assignment or a FV in each window. The new encoding needs a specific kernel (TFK) instead of standard kernels, such as linear, RBF, etc. Finally a multi-class SVM performs the recognition.

frame-windows on the video. The width of the window is w_K frames and it is displaced w_D frames each time.

A limitation may be introduced because of the width of the window, as the narrower the window the sparser the data used for encoding. In the case of BoF a descriptor is commonly assigned to the closest cluster which is a rough assignation because much of the spatial information in the descriptors space is lost. FV, on the other side, keeps information related to the mean and variance of each cluster which addresses this limitation. Soft-assignment has been proven a good improvement representing continuous data with a codebook model [19] and then in order to cope with the BoF limitation a soft-assignment is proposed. Specifically, first the relative distance between a descriptor S and a cluster centroid c_i in relation to the nearest cluster centroid is obtained.

$$\tilde{d}\left(S, c_i\right) = \frac{d\left(S, c_i\right)}{\min_j\left(d\left(S, c_j\right)\right)} \tag{1}$$

But, instead of performing a hard-assignment (one for the closest cluster and zero for the rest), a soft-assignment is applied as follows:

$$\tilde{s}\left(S, c_i\right) = \left(\frac{1}{\tilde{d}\left(S, c_i\right)}\right)^{\beta} \tag{2}$$

We assure that maximum value of $\tilde{s}\left(S, c_i\right) = 1$ is for the closest cluster while smaller values are assigned for more distant centroids. Also, high values of β approximate to the hard-assignment, which is achieved when $\beta \rightarrow \infty$.

Finally, we obtain a sequence $\mathbf{X} = \{\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_N\}$, being $\overline{\mathbf{x}}_i$ a D-dimensional vector. In the case of BoF D is the number of clusters in the codebook, while in the case of FV, where soft-assignment is not used, $D = K(2d+1)$, as discussed in section 2.

3.2. Time-Flexible Kernel (TFK)

In the standard approach we have a fixed size D-dimensional vector per video so only standard kernels (linear, polynomial, RBF, χ^2, ...) may be applied before using a multi-class SVM. In contrast, our encoding produces arbitrary length sequences of vectors and therefore our novel formulation of a kernel between sequences of different length is applied.

Having two sequences $\mathbf{X} = \{\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_N\}$ and $\mathbf{Y} = \{\overline{\mathbf{y}}_1, \ldots, \overline{\mathbf{y}}_M\}$ we define the function space $\Gamma : \mathbb{R} \longrightarrow \mathbb{R}^D$ where $F, G \in \Gamma$:

$$F(t) = \sum_{i=1}^{N} f_i(t)\overline{\mathbf{x}}_i \tag{3}$$

$$G(t) = \sum_{j=1}^{M} g_j(t)\overline{\mathbf{y}}_j \tag{4}$$

with $\overline{\mathbf{x}}_i, \overline{\mathbf{y}}_j \in \mathbb{R}^D$. We link each vector element with a specific function $f_i, g_j : \mathbb{R} \longrightarrow \mathbb{R}$ used to introduce the temporal position of each element. These functions weigh each sequence element according to variable t.

The TFK is then defined as:

$$TFK(F, G) = \int_t F\left(t\right)^T G\left(t\right) dt \tag{5}$$

With the aim of demonstrating that TFK is indeed a kernel we reorder the equation:

$$\begin{aligned} TFK(F, G) = \int_t \sum_{i=1}^{N} \left(f_i(t)\overline{\mathbf{x}}_i^T\right) \sum_{j=1}^{M} \left(g_j(t)\overline{\mathbf{y}}_j\right) dt = \\ \sum_{i=1}^{N}\sum_{j=1}^{M} \left(\int_t \left(f_i(t)\right)\left(g_j(t)\right) dt\right) \left(\left(\overline{\mathbf{x}}_i^T\right)\left(\overline{\mathbf{y}}_j\right)\right) = \\ \sum_{i=1}^{N}\sum_{j=1}^{M} K_{ST}\left(f_i\left(t\right), g_j\left(t\right)\right) K_{LIN}\left(\overline{\mathbf{x}}_i, \overline{\mathbf{y}}_j\right) \end{aligned} \tag{6}$$

To prove that Equation 6 represents a kernel, we follow several steps. First we check whether K_{ST} and K_{LIN} inside the summation are indeed kernels. The linear kernel, K_{LIN}, is well known. On the other hand, to assure that the structural kernel, K_{ST}, is a kernel we impose the following initial conditions on f_i and g_j: First, they should be square integrable, so $\int_t \left(f_i(t)\right)^2 dt$ and $\int_t \left(g_j(t)\right)^2 dt$

are well defined (not infinity). Second, $f_i(t), g_j(t) \geq 0, \forall t$. Thus, $f_i(t)$ and $g_j(t)$ belong to the Hilber-space L_2, hence the kernel is semi-positive definite [28].

We still need to prove that the summation of these kernels is a kernel. In this regard, we proceed with the following steps: First, we extend the vector representing the shortest sequence with zeros. Without loss of generality, let's assume that $N > M$, so we extend $\overline{\mathbf{y}}_j$ in such a way that $\overline{\mathbf{y}}_j = \overline{\mathbf{0}}$ for $j > M$ and we can use any function $g_j(t)$ for $j > M$ that fulfil the initial conditions. We also denote x_i^p the $p-th$ component of the $\overline{\mathbf{x}}_i$ vector, and the N-dimensional vector $\hat{\mathbf{x}}^p = (x_i^p, i = 1 \cdots N)$. Then, we develop the scalar product in Equation 6 as:

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{p} K\left(f_i\left(t\right), g_j\left(t\right)\right)(x_i^p)(y_j^p) = \sum_{p}((\hat{\mathbf{x}}^p))^T\mathbf{K}(\hat{\mathbf{y}}^p) \qquad (7)$$

where $\mathbf{K}$ is a $N \times N$ matrix $K_{i,j} = K\left(f_i\left(t\right), g_j\left(t\right)\right)$. The matrix $\mathbf{K}$ is a positive semidefinite matrix, since it corresponds to a kernel in the space of functions. Thus, each of the addends in Equation 7 is a kernel in a subspace, and the sum of kernels in all the subspaces is also a kernel in the global space [37].

As $\overline{\mathbf{x}}_i, \overline{\mathbf{y}}_j$ are vectors in an arbitrary $\mathbb{R}^D$ space, we can consider any projection of them in a different $\mathbb{R}^S$ space obtaining $\phi\left(\overline{\mathbf{x}}_i\right), \phi\left(\overline{\mathbf{y}}_j\right)$. Then, we can consider any kernel $K\left(\overline{\mathbf{x}}_i, \overline{\mathbf{y}}_j\right)$ as a linear kernel in the projected space $K_{LIN}\left(\phi\left(\overline{\mathbf{x}}_i\right), \phi\left(\overline{\mathbf{y}}_j\right)\right)$ so, in the previous proof, the linear kernel can be substituted by any arbitrary kernel.

3.3. Application of TFK in Activity Recognition

In a real world application there are video sequences with variable lengths, and the recording or segmentation of same event classes are not perfect and then they might start and end in different positions. This implies that when comparing two repetitions of the same activity class it is possible that only a portion of the sequence coincides. We can see this fact in Figure 4 where two sequences of the same activity class (somersault), extracted from the HMDB51 benchmark, coincide only in the final portions of the sequences.

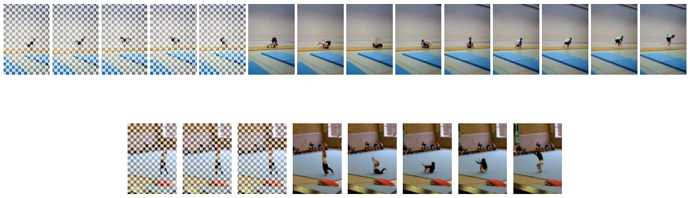


Figure 4: Activity correspondence: Two videos of somersault from HMDB51 where only the final portions of the sequences coincide.

Thanks to TFK we are able to compare sequences of different lengths and by selecting the appropriate associated function we can deal with non perfect alignment. In this regard we design the following framework.

Let $\mathbf{X} = \{\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_N\}$ and $\mathbf{Y} = \{\overline{\mathbf{y}}_1, \ldots, \overline{\mathbf{y}}_M\}$ be two sequences of vectors representing two different activity executions (same or different class). These sequences are obtained with the process explained in Section 3.1 so each vector of the sequence is a BoF or a FV. On the other hand, the proper alignment between two sequences is unknown and the computation of an algorithm seeking for this alignment can increase notably the computational cost. Moreover, a proper segmentation is assumed in advance so the core of the activity is most probably located in the middle of the sequences. Therefore, without an alignment process and simply ensuring that centres of both sequences coincide, the proposed method uses the structural kernel of TFK to provide the desired degree of flexibility in compression and stretching of the activity representation. As depicted in Figure 5, we center both sequences and assign a Gaussian distribution to each element of the sequences constrained to fixed temporal positions, being $f_i(t)$ and $g_j(t)$ the probability density functions of $\mathcal{N}_i$ and $\mathcal{N}_j$ respectively, $f_i(t) = \frac{1}{\sigma_x\sqrt{2\pi}} e^{-\frac{(t-\mu_i)^2}{2\sigma_x^2}}$ and $g_j(t) = \frac{1}{\sigma_y\sqrt{2\pi}} e^{-\frac{(t-\mu_j)^2}{2\sigma_y^2}}$ being $\mu_i = (i - \frac{N+1}{2})\Delta_t^x$ and $\mu_j = (j - \frac{M+1}{2})\Delta_t^y$.

The associated Gaussians weigh the inner product between sequence elements in relation to their temporal position, obtaining a maximum when their time coincides. These functions provide flexibility in the temporal position of

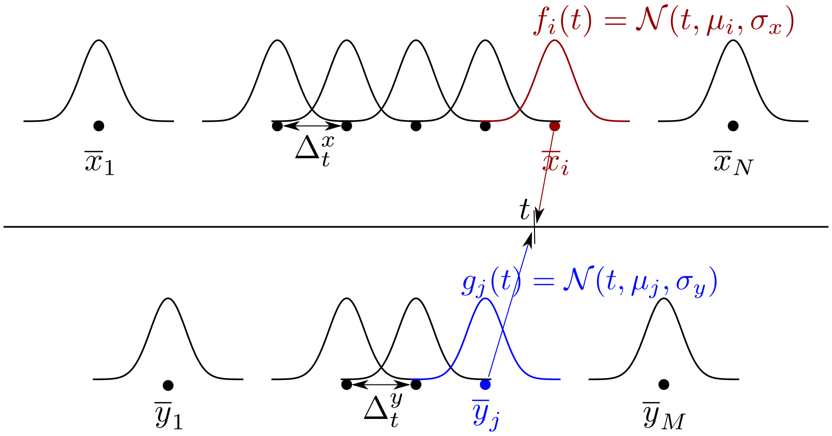


Figure 5: Kernel structure: Two centred sequences. Every vector of both sequences have a normal distribution associated. All the vector elements are compared in t weighted with their respective Gaussian function.

the elements, allowing an irregular expansion or narrowing of the sequences as well as a displacement. In order to define the functions $f_i(t)$ and $g_j(t)$ it is possible to fix the vector spacing Δ_t and then only the standard deviation of the Gaussian σ modifies the precision of the sequence position. The smaller is σ, the narrower are the Gaussians and then the lesser is the degree of temporal flexibility. Moreover, as the number of elements in a sequence is variable and each element has a Gaussian associated, it is possible to normalize the functions so that the length of the sequence does not influence the kernel value using the normalized Gaussians: $f'_i(t) = \frac{1}{N} f_i(t)$ and $g'_j(t) = \frac{1}{M} g_j(t)$.

Taking into account all previous concerns, we use the following kernel:

$$TFK(F,G) = \sum_{i=1}^{N}\sum_{j=1}^{M} K_{GAUSS_\rho}\left(f'_i(t), g'_j(t)\right) K_{LIN}\left(\overline{\mathbf{x}}_i, \overline{\mathbf{y}}_j\right) \quad (8)$$

We use the kernel between Gaussians that was proposed in [28] as K_{ST} in Equation 6, which in our one-dimensional case is simplified as:

$$K_{GAUSS_\rho}\left(f'_i(t), g'_j(t)\right) = \frac{(2\pi\sigma_x\sigma_y)^{(1-2\rho)/2}}{NM\sqrt{2\rho}} e^{\frac{-\|\mu_i-\mu_j\|^2}{4\sigma_x\sigma_y/\rho}} \quad (9)$$

Selecting $\rho = 1/2$ we obtain the Bhattacharyya kernel.

Inspired by the idea exposed in STPM [13], we explore the addition of two levels of granularity in the sequence division. Therefore, using a simple linear combination of kernels, that keeps the kernel property [38], we combine the TFK previously explained with a linear kernel between the vectors obtained from the feature extraction of the whole video. In Figure 3 this would mean to combine the two pipelines of the diagram in the following kernel.

$$CombK(v_1, v_2) = TFK(F, G) + K_{LIN}\left(\overline{\mathbf{x}}, \overline{\mathbf{y}}\right) \tag{10}$$

If the means and variances of the functions $f_i(t)$ and $g_j(t)$ are only dependant on the length of the sequences it is possible to precompute in advance the $K_{GAUSS_\rho}\left(f'_i(t), g'_j(t)\right)$ values for most of the possibles combinations of N and M, so the computational cost is only influenced by the kernel between the vectors. Considering the computational cost of $K\left(\overline{\mathbf{x}}_i, \overline{\mathbf{y}}_j\right)$ be $\mathcal{O}(D)$, the increase is linear with the increase of one of the sequences length $\mathcal{O}(NM(D+1))$.

4. Validation

4.1. Datasets

We test the performance of our framework in four challenging Activity-Recognition benchmarks (HMDB51, UCF50, OlympicSports, Virat Release 2.0), against other published methods.

HMDB51 [39] is one of the most challenging datasets nowadays. It contains a collection of videos obtained from a variety of sources ranging from digitized movies to YouTube videos. The total of 6766 video clips contains 51 distinct activity categories each one represented by at least 101 examples. The dataset is divided by the authors into 3 splits, each one containing 70 training clips and 30 testing clips in order to display a representative variability of the recording sources. The dataset includes a stabilized version of the videos that is not used in our experiments. We follow the protocol proposed by the authors.

UCF50 [40] is obtained from YouTube videos. It contains 6681 video clips of 50 different activities. Some of these videos are segmentations of a longer one, so it is important to follow the authors' protocol. The authors suggest a division into 25 groups in order to apply a leave-one-group-out cross-validation strategy that we follow.

OlympicSprots [36] contains 783 videos of athletes practising 16 different sports. All video sequences were obtained from YouTube and have been annotated with the help of Amazon Mechanical Turk. The authors suggest a split for training and testing the recognition system.

Virat Release 2.0 [41] has been recorded in 11 different scenes of video surveillance, captured by stationary HD cameras (1080p or 720p). There are 11 different classes of activities annotated where persons and vehicles appear (Loading, Unloading, Opening Trunk, Closing Trunk, Getting Into Vehicle, Getting out of Vehicle, Entering Facility, Exiting Facility, Gesturing, Carrying and Running). We follow the scene-independent learning and recognition mode of evaluation suggested by the authors. In the experiments we specifically focus in the actions with "opposite" counterpart, all the actions except Gesturing, Carrying and Running.

4.2. Parameter Identification

The framework performance is tested using two different descriptors. First, as representative low-level descriptor with short-term information we have used the MIP descriptor [5], which performs better than other common short-term descriptors like SIFT or HOG-HOF. Second, as state-of-the-art descriptor, and with longer temporal information captured we have selected the IDT descriptor. In order to understand the utility of the proposed framework we have conducted more exhaustive experiments with this descriptor. In both cases a parameter analysis is explained in this subsection.

The MIP descriptor is extracted with the original specifications proposed by the authors. The video is encoded with a BoF approach creating a codebook of 5000 codewords per channel obtaining a (8×5000)-dimensional vector. Our proposed video encoding depends on three parameters: β (in Equation 2) controlling the softness of the assignment and w_D and w_K (in Figure 3) modelling

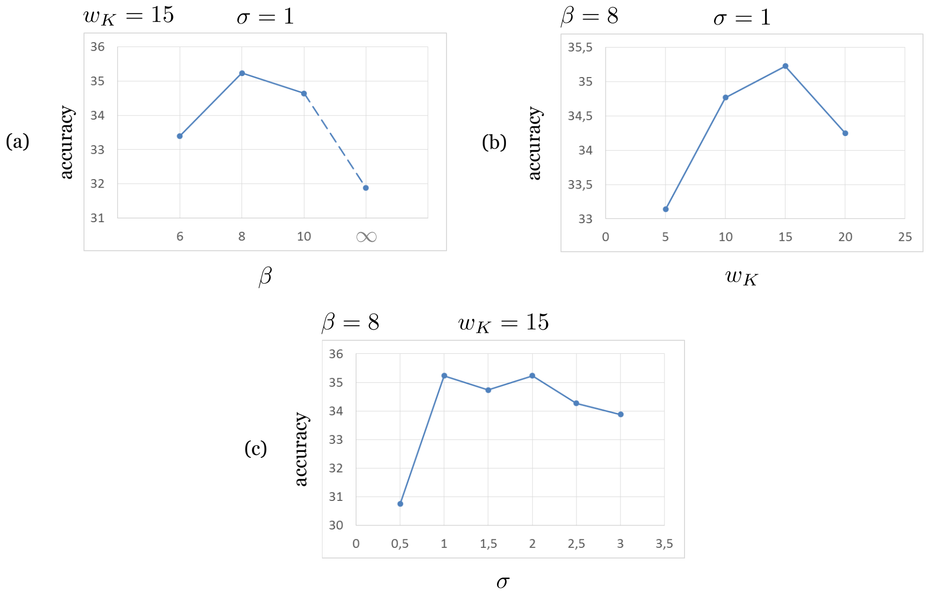


Figure 6: Parameters performance using MIP descriptors: Performance of the system evaluated in the first split of HMDB51 in relation to three parameters: (a) β, (b) w_K, and (c) σ.

the sliding frame-windows. Fixing the temporal spacing $\Delta_t = 1$ we let σ as the free parameter of the Gaussian functions.

We firstly perform experiments for multiple combinations of the parameter β, the window width w_D, the window displacement w_K and the the standard deviation σ of the Gaussian function of the kernel, using the first split of the HMDB51 dataset.

From initial experiments we have found that a sliding frame-window without overlapping provides best results, therefore we fix $w_D = w_K$ and then only 2 parameters of the video encoding are analysed: w_K and β. We show in Figure 6 the results obtained by fixing two of the three analysed parameters to the values finally selected, so the graphs represent the performance of the remaining one.

We can observe the importance of using the soft-assignment approximation in Figure 6(a) where different values of β are evaluated. If we use a hard-assignment with a window of width $w_K = 15$ the system performance declines in relation to a proper soft-assignment, which can be explained by the lack of sufficient data in a window. On the other hand, any of the three analysed values of β (6, 8 and 10) gives better results than the hard-assignment, what implies that the use of a soft-assignment is an adequate optimization in a wide range of values.

The width of the window w_K, Figure 6(b), does not impose significant variations in the system performance either, although the value $w_K = 15$ seems to be optimal.

Figure 6(c), depicts the performance of the system while varying the standard deviation σ of the kernel Gaussian functions. The bigger it is the wider are the Gaussians which means that the sequences are more flexible to asymmetrically expand or shrink but also that the temporal position is less influential. Very small values of σ lead to low accuracy, but then the variation in performance is minor and we find the optimum between $\sigma = 1$ and $\sigma = 2$.

We extend the parameters influence experiments to the IDT descriptor. In this case the encoding is performed with FV using a mixture model of 256 Gaussians. Hence, the study is performed over w_K and w_D for the sliding frame-window and σ for the temporal structure. We perform the analysis using the training examples of the OlympicSports dataset, dividing it into two groups randomly selected: 70% for training and 30% for validation. In Figure 7 we show three graphs fixing two of the parameters to the final value selected and varying the remaining one.

Figure 7 (a) shows low variation in the performance of the system for varying w_K except with short windows close to the IDT length, once reached $w_K = 25$ the accuracy variation is produced only by two examples recognition, slightly tending to the maximum when increasing the size until reaching the whole video represented in the axes as $w_K \to \infty$. The case where $w_K \to \infty$ corresponds to the use of a single FV per video, which is the standard approach of IDT presented in [6]. Therefore, in order to compare our framework with the original approach we carry out the experiments using $w_K = 30$. The value of w_D has even a lower influence in the performance and although the optimal value is

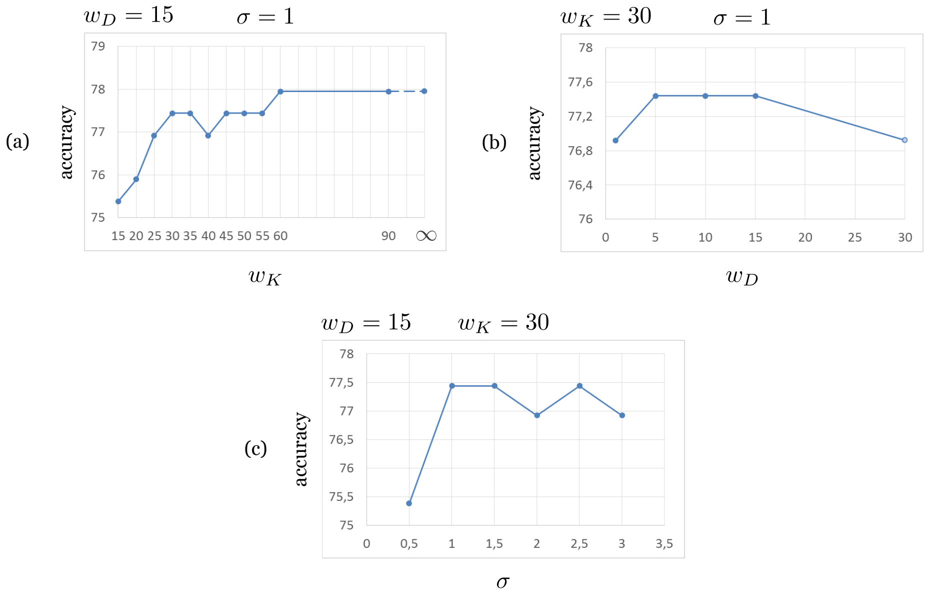


Figure 7: Parameters performance using IDT descriptors: Performance of the system evaluated in the OlympicSports dataset in relation to three parameters: (a) w_K, (b) w_D and (c) σ.

between $w_D = 5$ and $w_D = 15$, it makes little difference in the final result. The parameter σ has a similar behaviour to the one in the MIP analysis, although we can consider that the optimum value extends from $\sigma = 1$ to at least the maximum analysed $\sigma = 3$ because the accuracy decreases only in one incorrectly recognized example.

4.3. Framework Validation

Our proposed framework is advantageous in two scenarios: (i) when using short time descriptors, as it allows including longer temporal information in the classifier and (ii) recognizing complex activities where the order of actions may be crucial for correct recognition. In the latter scenario, our framework can benefit even cases where longer time descriptors are used, as it encodes all the temporal information of the activity. However, when dividing the videos in small sub-clips the information can be scarce and produce unstable encoding that leads to bad classification in examples where the previous propositions are not fulfilled. This problem is overcome by combining the two granularities of the video division proposed in Equation 10.

Our first experiment analysed the proposed framework performance using short-term descriptors. It is carried out using the MIP descriptor and the BoF encoding over the datasets HMDB51 and UCF50. In the previous section, we have seen that the performance of the framework is not significantly influenced when parameters are chosen within acceptable ranges. For the following experiments we select $\beta = 8$, in case of soft-assignment, the width $w_K = 15$ and the displacement of sliding frame-windows $w_D = 15$, so they are not overlapped, and the standard deviation of the Gaussian functions of the structural kernel $\sigma = 1$. We divided both datasets according to the authors' recommendations: 3 splits in HMDB51 and 25 groups in UCF50. We have used publicly available code for MIP [1] and SVM [2], using the default parameters. The randomness of initialisation of the k-means algorithm justifies why our results do not exactly coincide with those given in the MIP original paper. To ensure fair comparison between the standard method and our proposed framework, clustering and features extraction, as well as the one-against-all SVM classification, coincide in both pipelines. The difference lays in the middle stages. The standard method encodes the video with a single BoF obtained with a hard-assignment and applies a linear kernel between BoFs (BoF + LinK) as suggested by the authors for best results [5]. The proposed framework encodes each video in a sequence of BoFs (SeqBoF) using soft-assignment and applies the proposed TFK (SeqBoF + TFK). In Table 1 we can see how the inclusion of long-term temporal information using TFK clearly improves the results in both datasets which validates our first assumption regarding short-term descriptors. However, clearly better results have been obtained in the state-of-the-art using mid-term descriptors and specifically with IDT.

[1] MIP descriptor code can be downloaded in
http://www.openu.ac.il/home/hassner/projects/MIP/MIPcode.zip

[2] SVM code can be downloaded in http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

	HMDB51	UCF50
Kliper-Gross et al. [5]	29.17	68.51
MIP + BoF + LinK [5]	30.9	66.0
MIP + SeqBoF + TFK [OURS]	**34.4**	**72.4**

Table 1: Average accuracy (in %) in the HMDB51 and UCF50 datasets using the splits suggested by the authors. First row: results provided in [5]. Second row: own implementation of [5]. Third row: novel framework using TFK.

The second battery of experiments has been performed using the state-of-the-art descriptor, IDT[3], and comparing the novel framework against the original work in [6] and other related works in Activity Recognition. The sliding frame-windows vary form the MIP experiments as $w_K = 30$ and $w_D = 15$. We compute the experiments in all the evaluated datasets: OlympicSports, HMDB51, UCF50 and Virat Release 2. In all the experiments we follow the authors' recommendations: one division for training and testing in OlympicSports, 3 splits in HMDB51, leave-one-group-out from 25 groups in UCF50 and leave-one-scene-out with 11 scenes in Virat Release 2.0. As in the literature we find mainly results of Accuracy (acc) or Mean Average Precision (mAP), we compute both in the different approaches evaluated. Table 2 shows the results obtained with the original IDT as well as our proposed frameworks. To assure fair comparison we perform all the experiments using the same IDT extraction and GMM estimation. First, we obtain an unique FV per Video and apply a linear kernel for a SVM classification (IDT+FV+LinK), obtaining our own implementation of the approach in [6]. Following, we use the extracted IDT features to obtain a sequence of FV (SeqFV) that are used in our TFK approach (IDT+SeqFV+TFK). Finally, we combine both approaches in the CombK kernel (IDT+CombK).

The TFK approach is suitable in complex activities where the order of sub-

[3] IDT descriptor code can be downloaded in
http://lear.inrialpes.fr/people/wang/download/improved_trajectory_release.tar.gz

	OlympicSports		HMDB51		UCF50		Virat2	
	mAP	acc	mAP	acc	mAP	acc	mAP	acc
IDT+FV+LinK [6]	89.8	83.6	57.8	57.4	93.8	90.0	43.5	55.7
IDT+SeqFV+TFK	86.5	82.8	58.3	57.7	92.7	89.5	**52.1**	**63.6**
IDT+CombK	**89.9**	**84.3**	**59.1**	**58.6**	**94.1**	**90.3**	47.9	58.1

Table 2: Mean Average Precision (mAP) and average accuracy (acc) (in %) results in the OlympicSports, HMDB51, UCF50 and Virat Release 2 datasets using the Improved Dense Trajectories. First row: own implementation of [6]. Second row: novel framework using TFK. Third row: novel framework using the combination of TFK with [6].

actions determines the class. This can be confirmed with the results in Virat dataset where there are 4 activities with their respective “opposites”, depicted in the second row, two last columns of Table 2. However, TFK also have some drawbacks, as can be observed in the other three datasets where it performs similarly to the original IDT method. TFK relies in the extracted features in each window, and if they are scarce, the computed FV can be less robust to clutter. These datasets have complex activities, but not all of them depend on the order of sub-actions, therefore, although some activities are better classified with TFK, others are worse. The solution for this lack of robustness against clutter is the linear combination of both kernels. As we can see in the last row, this approach improves the results in all datasets but Virat where, even improving the original approach, the result is worse than the direct use of TFK because all the activities but 3 have “opposite” counterparts and the combination of kernels lowers the importance of order.

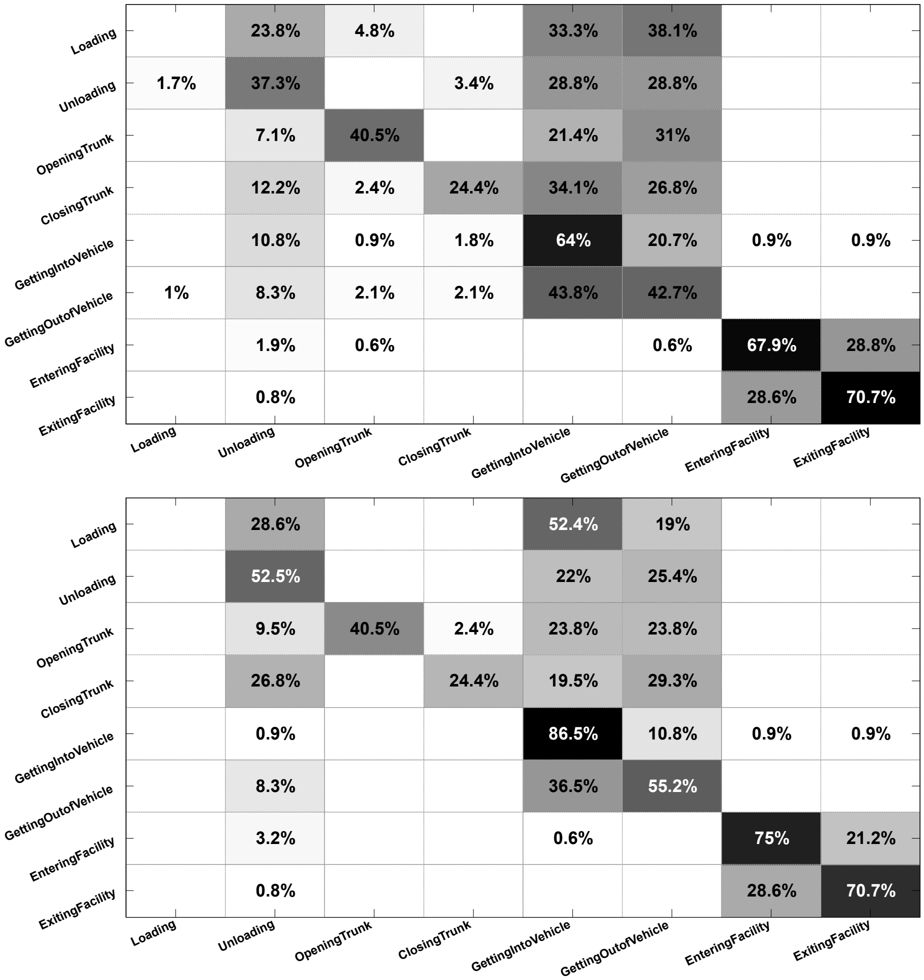
In Table 3 we observe a comparison of the proposed approach against some of the best results in the literature. It is worth to note that there are two rows representing the original approach with IDT [6], the third-to-last and the second-to-last. In the third-to-last row we show the results provided in the original paper, but a direct comparison between this results and our approach is not fair as several stages have some randomness and slightly alter the final results. On the other hand, the second-to-last row shows our own implementation of

	OlympicSports		HMDB51		UCF50		Virat2	
	mAP	acc	mAP	acc	mAP	acc	mAP	acc
Tang et al.[34]	66.8							
Niebles et al.[36]	72.1							
Li et al.[25]	76.2							
Gaidon et al. [12]		82.7						
Cao et al.[23]				27.8				
Kliper-Gross et al. [5]				29.2		68.5		
Reddy et al. [40]				27.0		76.9		
Wang et al. [6]	**90.2**			55.9		**90.5**		
IDT+FV+LinK [6]	89.8	83.6	57.8	57.4	93.8	90.0	43.5	55.7
IDT+CombK [OURS]	89.9	**84.3**	**59.1**	**58.6**	**94.1**	90.3	**47.9**	**58.1**

Table 3: Mean Average Precision (mAP) and average accuracy (acc) (in %) results in the OlympicSports, HMDB51, UCF50 and Virat Release 2 datasets. Comparison of the proposed framework (last row) with several state-of-the-art approaches.

[6] which share the features extraction and the clustering with the TFK so the comparison is fair. We can see how our novel approach overtakes all the compared methods in the "fair" comparison. To our knowledge there is no method with better results for all the datasets.

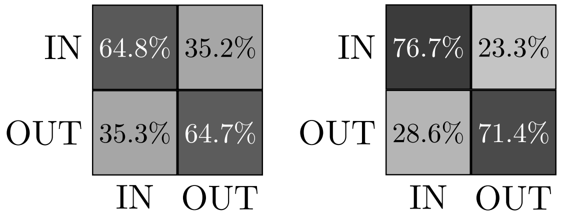
The results on Virat Release 2.0 are further analysed using only the activities with "opposites", (Loading, Unloading, Opening Trunk, Closing Trunk, Getting Into Vehicle, Getting out of Vehicle, Entering Facility and Exiting Facility). The Confusion Matrices of the (IDT+FV+LinK) and (IDT+SeqFV+TFK) methods are depicted in 8. In addition to the improvement obtained with the proposed framework, these matrices confirm our premise that the TFK is suitable for better learning of complex activities defined with the sub-actions order. The improvement achieved by the proposed framework is clear as every element of the diagonal is greater or equal. But the improvement do not restrict to this as in addition of a general improvement, the wrong classified activities are now confused with activities with similar temporal structure. For instance, the two first activities ('Loading' and 'Unloading') are mainly confused with ('Getting



	Loading	Unloading	OpeningTrunk	ClosingTrunk	GettingIntoVehicle	GettingOutofVehicle	EnteringFacility	ExitingFacility
Loading		23.8%	4.8%		33.3%	38.1%		
Unloading	1.7%	37.3%		3.4%	28.8%	28.8%		
OpeningTrunk		7.1%	40.5%		21.4%	31%		
ClosingTrunk		12.2%	2.4%	24.4%	34.1%	26.8%		
GettingIntoVehicle		10.8%	0.9%	1.8%	64%	20.7%	0.9%	0.9%
GettingOutofVehicle	1%	8.3%	2.1%	2.1%	43.8%	42.7%		
EnteringFacility		1.9%	0.6%			0.6%	67.9%	28.8%
ExitingFacility		0.8%					28.6%	70.7%

	Loading	Unloading	OpeningTrunk	ClosingTrunk	GettingIntoVehicle	GettingOutofVehicle	EnteringFacility	ExitingFacility
Loading		28.6%			52.4%	19%		
Unloading		52.5%			22%	25.4%		
OpeningTrunk		9.5%	40.5%	2.4%	23.8%	23.8%		
ClosingTrunk		26.8%		24.4%	19.5%	29.3%		
GettingIntoVehicle		0.9%			86.5%	10.8%	0.9%	0.9%
GettingOutofVehicle		8.3%			36.5%	55.2%		
EnteringFacility		3.2%			0.6%		75%	21.2%
ExitingFacility		0.8%					28.6%	70.7%

Figure 8: Confusion Matrices in Virat Release 2.0 using activities with "opposites": First the Confusion Matrix using IDT + FV + LinK approach, second using IDT + SeqFV + TFK.



	IN	OUT
IN	64.8%	35.2%
OUT	35.3%	64.7%

	IN	OUT
IN	76.7%	23.3%
OUT	28.6%	71.4%

Figure 9: Confusion Matrices in Virat Release 2.0 using two classes (IN and OUT): First the Confusion Matrix using IDT + FV + LinK approach, second using IDT + SeqFV + TFK.

Into Vehicle' and 'Getting Out of Vehicle'). If we observe the Confusion Matrix of the (IDT + FV + LinK) method, the confusion is more or less random, but using TFK we can see how the structure is learned and then "loading" is mainly confused with "getting into vehicle" and "unloading" is mainly confused with "getting out of vehicle". We achieve a clearer representation of this idea by gathering all the activities with similar temporal structure into one class so, activities (Loading, Opening Trunk, Getting Into Vehicle and Entering Facility) with structure (approaching and opening-closing) are grouped in class IN and activities (Unloading, Closing Trunk, Getting out of Vehicle and Exiting Facility) with structure (opening-closing and moving away) are grouped in class OUT. Figure 9 depicts the Confusion Matrices of these two classes. Here it is clear how the TFK approach keeps better the temporal structure of the activities.

Finally, we introduce one more experiment in order to compare our framework with the STPM approach which also preserves the temporal structure of the activities. In [13], *Choi et al.* have designed an experiment called *Quality of binary decision* where one example is compared to other two, one with the same class and other with a different class. Whenever the example of the same class is more similar to the initial example than the other one, the binary decision is correct. With this experimentation the authors obtains a maximum of 95.3% of Precision. In order to get a similar process we select single-class SVM to provide binary decisions between two randomly selected examples (one form the same class and one from a different class). Whenever the example of the same class has a greater note than the other one, the binary decision is correct. Using this experimentation we obtain a 99.3% of Precision.

5. Conclusion

We have introduced a new framework that improves accuracy in human activity classification taking into account the long-term information. The framework can be used with a wide variety of low-level feature descriptors, such as MIP and IDT, and video encoding methods, such as BoF and FV. The specific technical novelties of our work is a video encoding method that preserves the temporal information and the Time Flexible Kernel that is able to compare sequences of different lengths and random alignment.

Our experiments demonstrated the value of the novel framework in two cases: First, low-level descriptors with short-term information lose the long-term temporal information of the sequences. Our framework is able to consider such temporal information and therefore can improve the performance in activity recognition. Second, although modern state-of-the-art descriptors like IDT include some temporal information for recognizing several activities in spite of the unordered encoding of BoF or FV, they fail in case of complex activities that are defined by the order of the same short events. Again, our framework is able to preserve such complex temporal structure and distinguish between activities that consist of similar events but in different order.

The novel formulation of TFK is not restricted to activity sequences but it can be applied in any comparison between two sets that their structure of information can be defined using the functions f_i and g_j. For instance, an interesting extension for future research will be its application in image-based recognition, where the spacial structure is an important source of information.

Finally, the TFK approach can introduce some noisy results if the number of low-level extracted features is small in some windows. Using several levels of granularity in window width reduces this effect.

6. Acknowledgements

This work was partially supported by Spanish Grant TIN2013-45312-R (MINECO) and FEDER. Mario Rodriguez was sponsored by Spanish FPI Grant BES-2011-043752 and EEBB-I-14-08410.

References

[1] I. Laptev, On space-time interest points, International Journal of Computer Vision (IJCV) 64 (2005) 107–123. doi:10.1007/s11263-005-1838-7.

[2] P. Scovanner, S. Ali, M. Shah, A 3-dimensional sift descriptor and its application to action recognition, in: Proceedings of the 15th international conference on Multimedia, 2007, pp. 357–360. doi:10.1145/1291233.1291311.

[3] A. Kläser, M. Marszałek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, in: British Machine Vision Conference (BMVC), 2008, pp. 995–1004. doi:10.5244/C.22.99.

[4] D. Oneata, J. Verbeek, C. Schmid, Action and event recognition with fisher vectors on a compact feature set, in: IEEE Intenational Conference on Computer Vision (ICCV), 2013, pp. 1817–1824. doi:10.1109/ICCV.2013.228.

[5] O. Kliper-Gross, Y. Gurovich, T. Hassner, L. Wolf, Motion interchange patterns for action recognition in unconstrained videos, in: European Conference on Computer Vision (ECCV), 2012. doi:10.1007/978-3-642-33783-3_19.

[6] H. Wang, C. Schmid, Action recognition with improved trajectories, in: IEEE Intenational Conference on Computer Vision (ICCV), 2013. doi:10.1109/ICCV.2013.441.

[7] J. Yamato, J. Ohya, K. Ishii, Recognizing human action in time-sequential images using hidden markov model, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1992. doi:10.1109/CVPR.1992.223161.

[8] J. Wang, P. Liu, M. F. She, H. Liu, Human action categorization using conditional random field, in: Robotic Intelligence In Informationally Structured Space (RiiSS), 2011. doi:10.1109/RIISS.2011.5945793.

[9] Y.-M. Liang, S.-W. Shih, A. C.-C. Shih, H.-Y. M. Liao, C.-C. Lin, Learning atomic human actions using variable-length markov models, Transactions on Systems, Man and Cybernetics. Part B 39 (2009) 268–280. doi:10.1109/TSMCB.2008.2005643.

[10] R. Vezzani, D. Baltieri, R. Cucchiara, HMM based action recognition with projection histogram features, in: Recognizing Patterns in Signals, Speech, Images and Videos, Springer Berlin Heidelberg, 2010, pp. 286–293. doi:10.1007/978-3-642-17711-8_29.

[11] A. Antonucci, R. D. Rosa, A. Giusti, Action recognition by imprecise hidden markov models, in: International Conference on Image Processing, Computer Vision and Pattern Recognition (IPCV), 2011.

[12] A. Gaidon, Z. Harchaoui, C. Schmid, Recognizing activities with cluster-trees of tracklets, in: British Machine Vision Conference (BMVC), 2012, pp. 30.1–30.13. doi:10.5244/C.26.30.

[13] J. Choi, Z. Wang, S.-C. Lee, W. J. Jeon, A spatio-temporal pyramid matching for video retrieval, Computer Vision and Image Understanding 117 (6) (2013) 660 – 669. doi:http://dx.doi.org/10.1016/j.cviu.2013.02.003.

[14] I. Laptev, M. Marszałek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008. doi:10.1109/CVPR.2008.4587756.

[15] Q. V. Le, W. Y. Zou, S. Y. Yeung, A. Y. Ng, Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 2011, pp. 3361–3368. doi:10.1109/CVPR.2011.5995496.

[16] E. F. Can, R. Manmatha, Formulating action recognition as a ranking problem, in: International workshop on Action Similarity in Unconstrained Videos, 2013. doi:10.1109/CVPRW.2013.44.

[17] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, C.-W. Ngo, Trajectory-based modeling of human actions with motion reference points., in: European Conference on Computer Vision (ECCV), 2012. doi:10.1007/978-3-642-33715-4_31.

[18] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Dense trajectories and motion boundary descriptors for action recognition, International Journal of Computer Vision (IJCV) 103 (2013) 60–79. doi:10.1007/s11263-012-0594-8.

[19] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, J. M. Geusebroek, Visual word ambiguity, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (7) (2010) 1271–1283. doi:10.1109/TPAMI.2009.132.

[20] Y. Zhu, X. Zhao, Y. Fu, Y. Liu, Sparse coding on local spatial-temporal volumes for human action recognition, in: Asian Conference on Computer Vision (ACCV), 2011. doi:10.1007/978-3-642-19309-5_51.

[21] M. Rodriguez, C. Medrano, E. Herrero, C. Orrite, Transfer learning of human poses for action recognition, in: Human Behavior Unterstanding, 2013. doi:10.1007/978-3-319-02714-2_8.

[22] D. Xu, S.-F. Chang, Video event recognition using kernel methods with multilevel temporal alignment, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (11) (2008) 1985–1997. doi:10.1109/TPAMI.2008.129.

[23] L. Cao, Y. Mu, A. Natsev, S.-F. Chang, G. Hua, J. Smith, Scene aligned pooling for complex video recognition, in: European Conference on Computer Vision (ECCV), 2012. doi:10.1007/978-3-642-33709-3_49.

[24] A. Vahdat, K. Cannons, G. Mori, S. Oh, I. Kim, Compositional models for video event detection: A multiple kernel learning latent variable approach, in: IEEE Intenational Conference on Computer Vision (ICCV), 2013.

[25] W. Li, Q. Yu, A. Divakaran, N. Vasconcelos, Dynamic pooling for complex event recognition, in: IEEE International Conference on Computer Vision (ICCV), 2013. doi:10.1109/ICCV.2013.339.

[26] H. Shimodaira, K. ichi Noma, M. Nakai, S. Sagayama, Dynamic time-alignment kernel in support vector machine, in: Advances in Neural Information Processing Systems (NIPS), 2002.

[27] M. Cuturi, Fast global alignment kernels, in: International Conference on Machine Learning (ICML), 2011.
URL http://www.icml-2011.org/papers/489_icmlpaper.pdf

[28] T. Jebara, R. Kondor, A. Howard, Probability product kernels, Journal of Machine Learning Research 5 (2004) 819–844.
URL http://dl.acm.org/citation.cfm?id=1005332.1016786

[29] T. Jebara, Y. Song, K. Thadani, Spectral clustering and embedding with hidden markov models, in: European Conference on Machine Learning (ECML), 2007. doi:10.1007/978-3-540-74958-5_18.

[30] F. Zhou, F. De la Torre Frade, J. K. Hodgins, Hierarchical aligned cluster analysis for temporal clustering of human motion, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (2013) 582–596. doi:10.1109/TPAMI.2012.137.

[31] J. A. Rodriguez-Serrano, S. Singh, Trajectory clustering in CCTV traffic videos using probability product kernels with hidden markov mod-

els., Pattern Analysis & Applications 15 (2012) 415–426. doi:10.1007/s10044-012-0269-7.

[32] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, Vol. 2, 2006, pp. 2169–2178. doi:10.1109/CVPR.2006.68.

[33] M. Ryoo, J. Aggarwal, Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities, in: IEEE Intenational Conference on Computer Vision (ICCV), 2009, pp. 1593–1600. doi:10.1109/ICCV.2009.5459361.

[34] K. Tang, L. Fei-Fei, D. Koller, Learning latent temporal structure for complex event detection, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, 2012, pp. 1250–1257. doi:10.1109/CVPR.2012.6247808.

[35] S. Todorovic, Human activities as stochastic kronecker graphs, in: European Conference on Computer Vision (ECCV), 2012. doi:10.1007/978-3-642-33709-3_10.

[36] J. C. Niebles, C.-W. Chen, L. Fei-Fei, Modeling temporal structure of decomposable motion segments for activity classification, in: European Conference on Computer Vision (ECCV), 2010, pp. 392–405. doi:10.1007/978-3-642-15552-9_29.

[37] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[38] M. Gönen, E. Alpaydın, Multiple kernel learning algorithms, Journal of Machine Learning Research 12 (2011) 2211–2268.
URL http://www.jmlr.org/papers/v12/gonen11a.html

[39] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, HMDB: a large video database for human motion recognition, in: IEEE Intenational Conference on Computer Vision (ICCV), 2011. `doi:10.1109/ICCV.2011.6126543`.

[40] K. K. Reddy, M. Shah, Recognizing 50 human action categories of web videos, Machine Vision and Applications 24 (5) (2013) 971–981. `doi:10.1007/s00138-012-0450-4`.

[41] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, M. Desai, A large-scale benchmark dataset for event recognition in surveillance video, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 2011, pp. 3153–3160. `doi:10.1109/CVPR.2011.5995586`.