Action Recognition in Video Using Sparse Coding and Relative Features

Analí Alfaro
P. Universidad Catolica de Chile
Santiago, Chile
ajalfaro@uc.cl

Domingo Mery
P. Universidad Catolica de Chile
Santiago, Chile
dmery@ing.puc.cl

Alvaro Soto
P. Universidad Catolica de Chile
Santiago, Chile
asoto@ing.uc.cl

Abstract

This work presents an approach to category-based action recognition in video using sparse coding techniques. The proposed approach includes two main contributions: i) A new method to handle intra-class variations by decomposing each video into a reduced set of representative atomic action acts or key-sequences, and ii) A new video descriptor, ITRA: Inter-Temporal Relational Act Descriptor, that exploits the power of comparative reasoning to capture relative similarity relations among key-sequences. In terms of the method to obtain key-sequences, we introduce a loss function that, for each video, leads to the identification of a sparse set of representative key-frames capturing both, relevant particularities arising in the input video, as well as relevant generalities arising in the complete class collection. In terms of the method to obtain the ITRA descriptor, we introduce a novel scheme to quantify relative intra and inter-class similarities among local temporal patterns arising in the videos. The resulting ITRA descriptor demonstrates to be highly effective to discriminate among action categories. As a result, the proposed approach reaches remarkable action recognition performance on several popular benchmark datasets, outperforming alternative state-of-the-art techniques by a large margin.

1. Introduction

This work presents a new method for action recognition in video that incorporates two novel ideas: (1) A new method to select relevant key-frames from each video, and (2) A new method to extract an informative video descriptor. In terms of our technique for key-frame selection, previous works have also built their action recognition schemes on top of key-frames [45], Snippets [33], Exemplars [40], Actoms [14], or other informative subset of short video sub-sequences [27][30]. As a relevant advantage, by representing a video using a compressed set of distinctive subsequences, it is possible to eliminate irrelevant or noisy temporal patterns and to reduce computation, while still retaining enough information to recognize a target action [4]. Furthermore, it is possible to obtain a normalized video representation that avoids distracting sources of intra-class variation, such as different velocities in the execution of an action.

Previous works have mainly defined a set of key-frames using manual labelling [14], clustering techniques [45], or discriminative approaches [30]. In the case of clustering techniques, the usual loss functions produce a set of key-frames that captures temporal action patterns occurring frequently in the target classes. As a relevant drawback, training instances presenting less common patterns are usually poorly represented [47] and, as a consequence, the diversity of intra-class patterns is not fully captured. In the case of discriminative approaches, identification of relevant key-frames is usually connected to classification stages, focusing learning on mining patterns that capture relevant inter-class differences. As a relevant drawback, the mining of key-frames again does not focus directly on effectively capturing the diversity of intra-class patterns that usually arise in complex action videos.

In contrast to previous work, our method to select key-frames explicitly focuses on an effective mining of relevant intra-class variations. As a novel guiding strategy, our technique selects, from each training video, a set of key-frames that balances two main objectives: (i) They are informative about the target video, and (ii) They are informative about the complete set of videos in an action class. In other words, we simultaneously favour the selection of relevant particularities arising in the input video, as well as meaningful generalities arising in an entire class collection. To achieve this, we establish a loss function that selects from each video a sparse set of key-frames that minimizes the reconstruction error of the input video and the complete set of videos in the corresponding action class.

In terms of our technique to obtain an informative video descriptor, most current video descriptors are based on quantifying the absolute presence or absence of a set of visual features. Bag-of-Words schemes are a good example of this strategy [19]. As a relevant alternative, recent works have shown that the relative strength [42], or sim-

ilarity among visual features [29], can be a powerful cue to perform visual recognition. As an example, the work in [42] demonstrates a notable increase in object recognition performance by using the relative ordering, or rank, among feature dimensions. Similarly, the work in [21] achieves excellent results using a feature coding strategy based on similarities among pairs of attributes (similes).

In contrast to previous work, our method to obtain a video descriptor is based on quantifying relative intra and inter-class similarities among local temporal patterns or key-sequences. As a building block, we use our proposed technique to identify key-frames that are augmented with neighbouring frames to form local key-sequences encoding local action acts. These key-sequences, in conjunction with sparse coding techniques, are then used to learn temporal class-dependent dictionaries of local acts. As a key idea, cross-projections of acts into dictionaries coming from different temporal positions or action classes allow us to quantify relative local similarities among action categories. As we demonstrate, these similarities prove to be highly discriminative to perform action recognition in video.

In summary, our method initially represents an action in a video as a sparse set of meaningful local acts or key-sequences. Afterwards, we use these key-sequences to quantify relative local intra and inter-class similarities by projecting the key-sequences to a bank of dictionaries encoding patterns from different temporal positions or actions classes. These similarities form our core video descriptor that is then fed to a suitable classifier to access action recognition in video. Consequently, this work makes the following three main contributions:

- A new method to identify a set of relevant key-frames in a video that manages intra-class variations by preserving essential temporal intra-class patterns.
- A new method to obtain a video descriptor that quantifies relative local temporal similarities among local action acts.
- Empirical evidence indicating that the combination of the two previous contributions provides a substantial increase in action recognition performance with respect to alternative state-of-the-art techniques.

2. Related Works

There is a large list of works related to category-based action recognition in video, we refer the reader to [1] for a suitable review. Here, we focus our review on methods that also decompose the input video into key-sequences, propose related video descriptors, or use sparse coding.

Key-sequences: Several previous works have tackled the problem of action recognition in video by representing each video by a reduced set of meaningful temporal parts. Weiland and Boyer [40] propose an action recognition approach based on key-frames that they refer to as Exemplars. Schindler and Van Gool [33] add motion cues by studying the amount of frames, or Snippets, needed to recognize periodic human actions. Gaidon et al. [14] present an action recognition approach that is built on top of atomic action units, or Actoms. As a relevant disadvantage, at training time, these previous methods require a manual selection or labelling of a set of key-frames or key-sequences.

Discriminative approaches to identify key-frames have also been used. Zhao and Elgammal [45] use an entropy-based score to select as key-frames the most discriminative frames from each video. Liu et al. [26] propose a method to select key-frames using the Adaboost classifier to identify highly discriminative frames for each target class. Extending DPMs [12] to action recognition, Niebles et al. [27] represent a video using global information and short temporal motion segments. Raptis and Sigal [31] use a video frame representation based on max-pooled Poselet [7] activations, in conjunction with a latent SVM approach to select relevant key-frames and learn action classifiers. In contrast to these previous approaches, we do not assume that all videos in an action class share a common set of key-sequences. In our case, we adaptively identify in each video key-sequences that consider reconstruction error and similarities to other local temporal patterns present in the class collection.

Video descriptors: Extensive research has been oriented to propose suitable spatio-temporal low-level features [22, 9, 20, 23, 24, 44, 37]. In our case, we build our descriptor on top of key-sequences that are characterized by low-level spatio-temporal features. In this sense, the proposed descriptor is more closely related to mid-level representations, such as the ones described in [25, 32, 39]. In contrast to our approach, current mid-level representations do not encode local temporal similarities among key-sequences.

In terms of encoding similarities among training instances, Kumar et al. [21] propose a method that exploits facial similarities with respect to a specific list of reference people. Yagnik et al. [42] presents a locality sensitive hashing approach that provides a feature representation based on relative rank ordering. Similarly, Parikh and Grauman [29] use a max-margin approach to learn a function that encodes relative rank ordering. Wang et al. [36] present a method that uses information about object-class similarities to train a classifier that responds more strongly to examples of similar categories than to examples of dissimilar categories. These previous works share with our approach the idea of explicitly encoding the relative strength of visual properties to achieve visual recognition. However, they are not based on sparse coding, or they do not exploit relative temporal relations among visual patterns.

Sparse Coding: A functional approach to action recognition is to create dictionaries based on low-level representations. Several methods can be used to produce a suitable dictionary, BoW [14, 23, 24], Fisher vectors [28, 5], random

forest [43, 44], and sparse coding techniques [16, 17, 35, 8]. Tran et al. [35] use motion information from human body parts and sparse coding techniques to classify human actions in video. For each body part, they build a dictionary that integrates information from all classes. Similarly to our approach, the atoms in each dictionary are given by the training samples themselves. As a main drawback, at training time, this method requires manual annotation of human body parts. Guha and Ward [16] explore several schemes to construct an overcomplete dictionary from a set of spatio-temporal descriptors extracted from training videos, however, this method does not use key-sequences or relative features in its operation. Castrodad et al. [8] propose a hierarchical two-level sparse coding approach for action recognition. In contrast to our approach, this work uses a global representation that discards local temporal information. Furthermore, it does not exploit key-frames or intra-class relations.

3. Our Method

Our proposed method has three main parts: i) Video Decomposition, ii) Video Description, and iii) Video Classification. We explain next the main details behind each of these parts.

3.1. Video Decomposition

Fig. 1 summarizes the main steps to decompose an input video into a set of K key-sequences. We explain next the details.

3.1.1 Selection of Key-Frames

We address the selection of key-frames from an action video as a reconstruction problem using sparse coding techniques [10]. Let $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^{p}$ be a set of p training videos of a given action class, where video $\mathbf{v}_i$ contains n_i frames $\mathbf{f}_i^j$, $j \in [1 \dots n_i]$. We encode each frame $\mathbf{f}_i^j$ using a pyramid of histograms of oriented gradients or PHOG–descriptor [6]. Then, video $\mathbf{v}_i$ is represented by a matrix $\mathbf{Z}_i \in \mathbb{R}^{m \times n_i}$, where column j contains the m-dimensional PHOG–descriptor of frame $\mathbf{f}_i^j$.

Our sparse coding representation considers two main design goals. First, similarly to [11], the atoms of the resulting representation must correspond to frames from the input video. Second, as mentioned before, the resulting atoms must simultaneously provide a suitable representation of the input video and the complete class. To achieve this, for each input video we solve the following optimization:

$$\min_{\mathbf{W}_i, \mathbf{W}_{(-i)}} \|\mathbf{Z}_i - \mathbf{Z}_i\mathbf{W}_i\|_F^2 + \alpha\|\mathbf{Z}_{(-i)} - \mathbf{Z}_i\mathbf{W}_{(-i)}\|_F^2 \quad (1)$$

$$\text{s.t. } \|\mathbf{W}_i\|_{1,2} \leq \lambda,$$
$$\|\mathbf{W}_{(-i)}\|_{1,2} \leq \lambda,$$

where $\mathbf{W}_i \in \mathbb{R}^{n_i \times n_i}$ corresponds to the matrix of coefficients that minimize the constrained reconstruction of the n_i frame descriptors in $\mathbf{Z}_i$. $\mathbf{Z}_{(-i)} = [\dots, \mathbf{Z}_{i-1}, \mathbf{Z}_{i+1}, \dots] \in \mathbb{R}^{m \times (n-n_i)}$ corresponds to the matrix of PHOG descriptors for all the n frames in a target class, excluding the n_i frames from video $\mathbf{v}_i$. $\mathbf{W}_{(-i)} = [\dots, \mathbf{W}_{i-1}, \mathbf{W}_{i+1}, \dots] \in \mathbb{R}^{n_i \times (n-n_i)}$ corresponds to the sparse representation of $\mathbf{Z}_{(-i)}$ using the frame descriptors in $\mathbf{Z}_i$. The mixed ℓ_1/ℓ_2 norm is defined as $\|\mathbf{A}\|_{1,2} \triangleq \sum_{i=1}^{N} \|\mathbf{a}_i\|_2$, where $\mathbf{A}$ is a sparse matrix and $\mathbf{a}_i$ denotes the i-th row of $\mathbf{A}$. Then, the mixed norm expresses the sum of the ℓ_2 norms of the rows of $\mathbf{A}$. Parameter $\lambda > 0$ controls the level of sparsity in the reconstruction, and parameter $\alpha > 0$ balances the penalty between errors in the reconstruction of video $\mathbf{v}_i$ and errors in the reconstruction of the remaining videos in the class collection. Following [11], we solve the constrained optimization in Eq. 1 using the *Alternating Direction Method of Multipliers (ADMM)* technique [13].

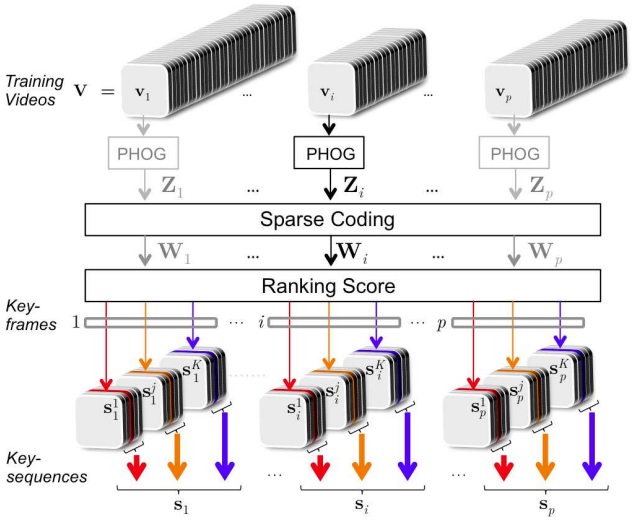

Figure 1. Overview of the proposed method to extract key-sequences from an input video.

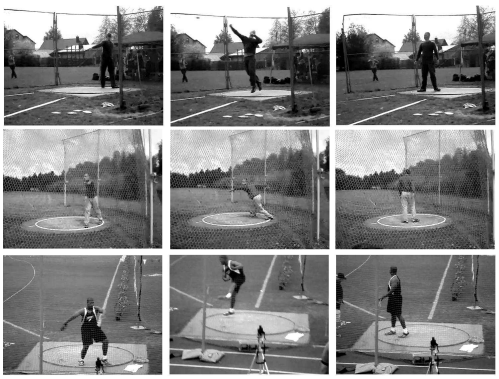

Figure 2. Key-frames selected by the proposed method (rows) for videos of the action category *Discus throwing* in the Olympic dataset using K=3.

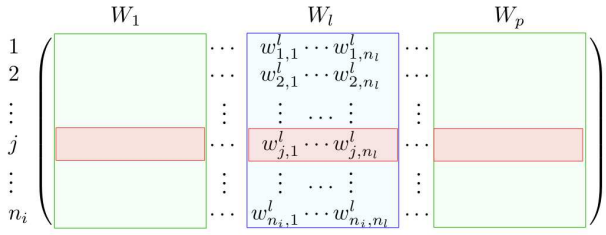

Figure 3. Matrix $\mathbf{W^i}$. Columns of $\mathbf{W^i}$ can be decomposed according to the p videos in an action class: $\mathbf{W^i} = [W_1, \cdots, W_p]$. This decomposition highlights that each row j in a submatrix W_l contains information about the contribution delivered by frame j in video $\mathbf{v}_i$ to reconstruct the frames in video $\mathbf{v}_l$. Then, row j in matrix $\mathbf{W^i}$ contains all the reconstruction coefficients associated to frame j in video v_i.

3.1.2 Selection of Key-Sequences

Matrix $\mathbf{W^i} = [\mathbf{W}_i|\mathbf{W}_{(-i)}]$ provides information about the contribution of each frame in $\mathbf{v}_i$ to summarize each of the videos in the entire class collection. Fig. 3 shows a diagram of matrix $\mathbf{W^i}$ that highlights this property. Specifically, each row j in $\mathbf{W^i}$ provides information about the contribution provided by frame j in video $\mathbf{v}_i$, $\mathbf{f}_i^j$, to reconstruct the p videos in the class collection. Using this property and the notation in Fig. 3, we define the following score to quantify the contribution of frame $\mathbf{f}_i^j$ to the reconstruction process:

$$\mathrm{R}(\mathbf{f}_i^j) = \sum_{l=1}^{p} \sum_{s=1}^{n_l} w^l_{j,s}. \tag{2}$$

$\mathrm{R}(\mathbf{f}_i^j)$ corresponds to the sum of the elements in the j-th row of matrix $\mathbf{W^i}$. We use this score to rank the frames in video $\mathbf{v}_i$ according to their contribution to the reconstruction process. In particular, a frame with a high ranking score provides a high contribution to the reconstruction of the videos in the class collection. Therefore, high scoring frames represent good candidates to be selected as key-frames for video $\mathbf{v}_i$.

Let $\mathbf{L}_i$ be the set of frames $\mathbf{f}_i^j$ from $\mathbf{v}_i$ that satisfy $\mathrm{R}(\mathbf{f}_i^j) > \theta$, where θ is a given threshold. We obtain a set of key-frames from $\mathbf{v}_i$ by selecting K frames from the candidates in $\mathbf{L}_i$. Several criterion can be used to select these K frames. In particular, to guarantee that the selected key-frames provide a good temporal coverage of the input video, we use the following scheme. First, we select K time instants uniformly distributed with respect to the length of the video. Then, for each of these time instants, we select as a key-frame the closest neighbouring frame in $\mathbf{L}_i$. Fig. 2 shows instances of key-frames selected by this approach using $K = 3$.

To include motion cues, we add neighbouring frames to each key-frame in order to form brief video acts that we refer to as key-sequences. Specifically, for a key-frame $\mathbf{f}_i^j$ in video $\mathbf{v}_i$, its corresponding key-sequence is given by the set $\mathbf{s}_i^j = \{\mathbf{f}_i^l\}_{l=j-t}^{l=j+t}$, i.e., $2t+1$ consecutive frames centered at the corresponding key-frame ($t \in \mathbb{N}$). Consequently, each input video $\mathbf{v}_i$ is decomposed into a set $\mathbf{s}_i = \{\mathbf{s}_i^1, \ldots, \mathbf{s}_i^K\}$, corresponding to K temporally ordered key-sequences.

3.2. Video Description

Fig. 4 summarizes the main steps to build our video descriptor. We explain next the details.

3.2.1 Relative Local Temporal Features

At the core of our method to obtain relative features is the use of sparse coding to learn a set of dictionaries that encode local temporal patterns present in the action classes. Specifically, in the case of C action classes and K temporal key-sequences, we use training data to learn a total of $C \times K$ dictionaries, where dictionary $D_{k_j}^{c_i}$, $c_i \in [1 \ldots C]$, $k_j \in [1 \ldots K]$, encodes relevant local temporal patterns occurring in class c_i at time instance k_j.

As a key observation, by projecting a given key-sequence to a concatenated version of a subset of the dictionaries, it is possible to quantify the relative similarity between the key-sequence and the individual dictionaries. This can be achieved by quantifying the total contribution of the atoms in each individual dictionary to the reconstruction of the projected key-sequence. As an example, consider the case of a concatenated dictionary that encodes local patterns learnt from sport actions. In this case, key-sequences from an action class such as *running* should use in their reconstruction a significant amount of dictionary atoms coming from similar action classes, such as *jogging* and *walking*. As a consequence, by quantifying the cross-talk among reconstruction contributions coming from different dictionaries, one can obtain a feature vector that encodes relative local similarities between the projected key-sequence and the temporal patterns encoded in each dictionary. Next, we exploit this property to apply two concatenation strategies that allow us to obtain a video descriptor capturing inter and intra-class similarity relations.

3.2.2 Inter-class Relative Act Descriptor

Our method to obtain inter-class relative local temporal features is composed of three main steps. In the first step we obtain a low-level feature representation for each key-sequence. Specifically, we randomly sample a set of spatio-temporal cuboids (300 in our experiments) from each key-sequence. These cuboids are encoded using the spatio-temporal HOG3D descriptor [20]. Section 4 provides further implementation details.

In the second step we use the resulting HOG3D descriptors and sparse coding to build a set of local temporal dictionaries for each class. Temporal locality is given by organizing the key-sequences according to their K temporal positions in the training videos. Let $\mathbf{Y}_j^c$ be the set of HOG3D

descriptors extracted from all key-sequences occurring at the j-th temporal position in the training videos from class c, where $j \in [1, \dots, K]$, $c \in [1, \dots, C]$. We find a class-based temporal dictionary $\mathbf{D}_j^c$ for position j using the K-SVD algorithm [2] to solve:

$$\min_{\mathbf{D}_j^c, \mathbf{X}_j^c} \|\mathbf{Y}_j^c - \mathbf{D}_j^c \mathbf{X}_j^c\|_{\mathsf{F}}^2 \quad \text{s.t.} \quad \|\mathbf{x}_i\|_0 \leq \lambda_1, \tag{3}$$

where $\mathbf{Y}_j^c \in \mathbb{R}^{m \times n_s}$, m is the dimensionality of the descriptors and n_s is the total number of cuboids sampled from videos of class c and temporal position j, $\mathbf{D}_j^c \in \mathbb{R}^{m \times n_a}$, $\mathbf{X}_j^c \in \mathbb{R}^{n_a \times n_s}$, n_a is the number of atoms in each dictionary $\mathbf{D}_j^c$, and the sparsity restriction on each column $\mathbf{x}_i \in \mathbf{X}_j^c$ indicates that its total number of nonzero entries must not exceed λ_1.

Finally, in the third step we use the previous set of dictionaries to obtain a local temporal similarity descriptor for each key-sequence. To achieve this, for each temporal position j, we concatenate the C class-based dictionaries obtained in the previous step. This provides a set of K temporal dictionaries, where each dictionary contains information about local patterns occurring in all target classes at a given temporal position j. These K representations allow us to quantify local temporal similarities among the target classes. Specifically, let $\mathbf{D}_j = [\mathbf{D}_j^1 \ \mathbf{D}_j^2 \ \dots \ \mathbf{D}_j^C]$ be the concatenated temporal dictionary corresponding to temporal position j. To obtain a descriptor for key-sequence $\mathbf{s}_i^j$ from video $\mathbf{v}_i$, we first project $\mathbf{s}_i^j$ onto dictionary $\mathbf{D}_j$ imposing a sparsity constraint. We achieve this by using the Orthogonal Matching Pursuit (OMP) technique to solve:

$$\min_{\mathbf{x}_i^j} \|\mathbf{s}_i^j - \mathbf{D}_j \mathbf{x}_i^j\|_{\mathsf{F}}^2 \quad \text{s.t.} \quad \|\mathbf{x}_i^j\|_0 \leq \lambda_2, \tag{4}$$

where vector $\mathbf{x}_i^j = \{\mathbf{x}_i^j[\mathbf{D}_j^1], \dots, \mathbf{x}_i^j[\mathbf{D}_j^C]\}$ is the resulting set of coefficients, and a component vector $\mathbf{x}_i^j[\mathbf{D}_j^c] \in \mathbb{R}^{n_a}$ corresponds to the coefficients associated to the projection of $\mathbf{s}_i^j$ onto the atoms in subdictionary $\mathbf{D}_j^c$.

We quantify the similarity of $\mathbf{s}_i^j$ to the atoms corresponding to each class by using a sum-pooling operator that evaluates the contribution provided by the words in each subdictionary $\mathbf{D}_j^c$ to the reconstruction of $\mathbf{s}_i^j$. We define this sum-pooling operator as:

$$\phi_j^c(\mathbf{s}_i^j) = \sum_{l=1}^{n_a} \mathbf{x}_i^j[\mathbf{D}_j^c](l). \tag{5}$$

By applying the previous method to the set of K key-sequences $\mathbf{s}_i^j$ in a video $\mathbf{v}_i$, we obtain a video descriptor $\mathbf{\Phi}^i = [\phi^1, \dots, \phi^K] \in \mathbb{R}^{C \times K}$, where each component vector ϕ^j is given by $\phi^j = [\phi_j^1, \dots, \phi_j^C]$. In this way, $\mathbf{\Phi}^i$ contains information about relative *inter-class* similarities among key-sequences or acts. Therefore, we refer to this descriptor as *Inter-class Relative Act Descriptor*.

3.2.3 Intra-class Relative Act Descriptor

The procedure in Section 3.2.2 provides a descriptor that encodes relative local temporal similarities across the target classes. In this section, we use a similar procedure to obtain local temporal similarities at an intra-class level. Specifically, we quantify the similarity of a key-sequence occurring at temporal position j with respect to the patterns occurring at the remaining $K-1$ temporal positions in a target class. To do this, we follow the procedure described in Section 3.2.2 but, this time we project a key-sequence $\mathbf{s}_i^j$ onto the concatenated dictionary $\mathbf{D}_{(-j)}^c = [\dots, \mathbf{D}_{j-1}^c \ \mathbf{D}_{j+1}^c \ \dots]$, i.e., the concatenation of the $k-1$ key-sequence dictionaries for class c, excepting the dictionary corresponding to temporal position j. We again use the OMP technique to perform this projection, i.e., to solve:

$$\min_{\mathbf{x}_i^j} \|\mathbf{s}_i^j - \mathbf{D}_{(-j)}^c \mathbf{x}_i^j\|_{\mathsf{F}}^2 \quad \text{s.t.} \quad \|\mathbf{x}_i^j\|_0 \leq \lambda_3. \tag{6}$$

Similarly to the Inter-class Relative Act Descriptor, we obtain a video descriptor, $\mathbf{\Psi}^i = [\psi^1, \dots, \psi^K] \in \mathbb{R}^{K \times (K-1)}$, by applying the projection to all key-sequences in a video $\mathbf{v}_i$ and then using the corresponding sum-pooling operations to quantify the reconstruction contribution of each subdictionary $\mathbf{D}_l^c$. In this way, $\mathbf{\Psi}^i$ contains information about relative *intra-class* similarities among key-sequences or local acts, therefore, we refer to this descriptor as *Intra-class Relative Act Descriptor*.

3.2.4 Inter Temporal Relational Act Descriptor: ITRA

We obtain a final feature vector descriptor for a video $\mathbf{v}_i$ by concatenating the Inter and Intra-class Relative Act Descriptors. We refer to this new descriptor as *Inter Temporal Relational Act Descriptor* or ITRA, where *ITRA*$(\mathbf{v}_i)$ = $\{\mathbf{\Phi}^i \cup \mathbf{\Psi}^i\} \in \mathbb{R}^{K \times (C + (K-1))}$.

3.3. Video Classification

ITRA can be used by any off-the-shelf supervised classification scheme. Here, we use a sparse coding approach. **Training:** During the training phase, we first use the method described in Section 3.1 to decompose each training video into a set of key-sequences. Then, we use the method described in Section 3.2 to obtain the ITRA descriptor for each training video. Afterwards, these descriptors, along with sparse coding techniques, are used to build a dictionary for each target class. Specifically, let $\mathbf{Y}^c$ be a matrix containing in its columns the ITRA descriptors corresponding to the training videos from action class $c \in [1, \dots, C]$. For each action class, we use the K-SVD algorithm [2] to obtain a class-based dictionary $\mathbf{B}^c$ by solving:

$$\min_{\mathbf{B}^c, \mathbf{X}^c} \|\mathbf{Y}^c - \mathbf{B}^c \mathbf{X}^c\|_{\mathsf{F}}^2 \quad \text{s.t.} \quad \|\mathbf{x}_i\|_0 \leq \lambda_4, \ \ \forall i, \tag{7}$$

where $\mathbf{B}^c \in \mathbb{R}^{|\text{ITRA}| \times n_a}$, $|\text{ITRA}|$ represents the dimensionality of the ITRA descriptor, and n_a is the selected number of

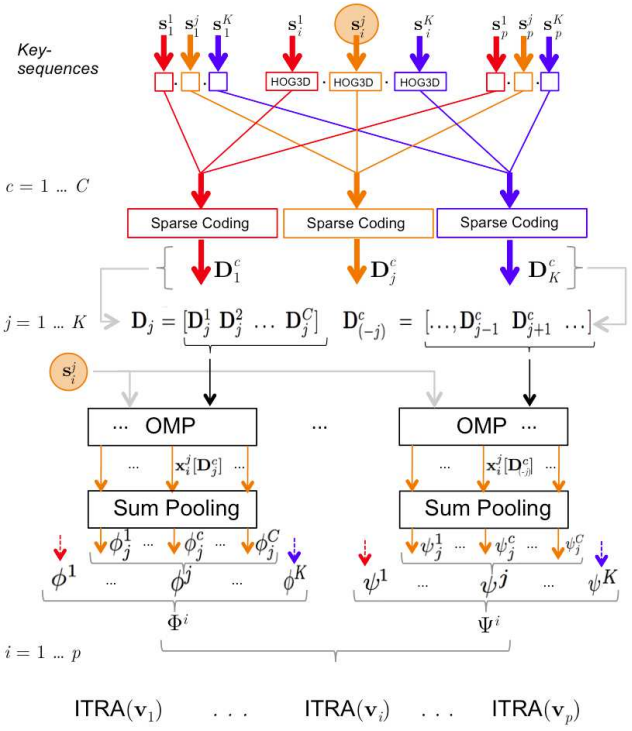

Figure 4. Overview of the method to obtain the ITRA descriptor. See Section 3.2 for details.

atoms to build the dictionary. $\mathbf{X}^c$ corresponds to the matrix of coefficients and vectors $\mathbf{x}_i$ to its columns. As a final step, we concatenate the C class-based dictionaries to obtain the joint dictionary $\mathbf{B} = [\mathbf{B}^1|\mathbf{B}^2|\cdots|\mathbf{B}^C]$ that forms the core representation to classify new action videos.

Inference: To classify a new input video, similarly to the training phase, we first use the methods in Sections 3.1 and 3.2 to obtain its ITRA descriptor. As a relevant difference from the training phase, in this case we do not know the class label of the input video, therefore, we need to obtain its key-sequence decomposition with respect to each target class. This task leads to C ITRA descriptors to represent each input video. Consequently, the classification of an input video consists of projecting these C descriptors onto the joint dictionary $\mathbf{B}$ and then using a majority vote scheme to assign the video to the class that contributes the most to the reconstruction of the descriptors. Specifically, let $\mathbf{v}_q$ be a test video, and $\Omega^c(\mathbf{v}_q)$ its ITRA descriptor with respect to class c. We obtain a sparse representation for each of the C ITRA descriptors using the OMP technique to solve:

$$\min_{\alpha_c} \|\Omega^c(\mathbf{v}_q) - \mathbf{B}\alpha_c\|_2^2 \quad \text{s.t.} \quad \|\alpha_c\|_0 \leq \lambda_5. \qquad (8)$$

The previous process provides C sets of coefficients α_c. We use each of these sets to obtain a partial classification of the input video. We achieve this by applying, to each set, a sum-pooling operator similar to the one presented in Eq. (5), and classifying the input video according to the class that presents the greatest contribution to the reconstruction of the corresponding set α_c. Finally, using these C partial classifications, we use majority vote to assign the input video to the most voted class.

4. Experiments and Results

We validate our method by using three popular benchmark datasets for action recognition: KTH [34], Olympic [27], and HOHA [23]. In all the experiments, we select values for the main parameters using the following criteria.

Estimation of Key-Sequences: We use training data from the Olympic dataset to tune the number of acts to represent an action. Experimentally, we find that 3 acts are enough to achieve high recognition performance. Hence, in all our experiments, we select $K = 3$ key-sequences to represent each training or test video.

In terms of the time span of each key-sequence, we take a fixed group of 7 frames ($t = 3$) to form each key-sequence. For each sequence we randomly extract 300 cuboids, described using HOG3D (300 dimensions). To filter out uninformative cuboids, we set a threshold to the magnitude of the HOG3D descriptor. We calibrate this threshold to eliminate the 5% least informative cuboids from each dataset. Afterwards, the remaining descriptors are normalized. Table 1 shows the value of the resulting thresholds for each dataset.

Dataset	Train	Test
KTH	2.5	2.5
Olympic	2	2
HOHA	1.3	1.6

Table 1. Thresholds used to filter out uninformative cuboids from the key-sequences. For each dataset, we calibrate this threshold to eliminate the 5% least informative cuboids.

Estimation of ITRA descriptor: Parameters for the extraction of ITRA descriptors are related to the construction of the dictionaries described in Section 3.2. Let μ be the redundancy [1] and let δ be the dimensionality of a descriptor. Following the empirical results in [16], we fix the number of atoms in each local dictionary to be $n_a = \mu \times \delta$. Therefore, the number of atoms for the concatenated dictionaries are: $P = \mu \times \delta \times C$ for the extraction of *Inter-class Relative Act Descriptors*, $\mathbf{\Phi}$, and $P = \mu \times \delta \times (K-1)$ for the extraction of *Intra-class Relative Act Descriptors*, $\mathbf{\Psi}$. In our experiments, we use $\mu = 2$ and $\delta = 300$. As a result, the dimension of the ITRA descriptors for KTH, Olympic, and HOHA datasets are 24, 54 and 30, respectively. Also, following [16], the sparsity parameters λ_1, λ_2, and λ_3, are set

[1] Redundancy indicates the folds of basis vectors that need to be identified with respect to the dimensionality of the descriptor.

to be 10% of the number of atoms.
Classifier: Parameters P, μ, λ_4, and λ_5 are configured using the same scheme described above.

4.1. Action Recognition Performance

KTH Dataset: This set contains 2391 video sequences displaying six types of human actions. In our experiments we use the original setup [34] to divide the data into training and test sets. Table 2 shows the recognition performance reached by our method. Table 2 also includes the performance of alternative action recognition schemes proposed in the literature, including approaches that also use sparse coding techniques [3, 8]. Our method obtains a recognition performance of 97.5%.

Method	Acc.
Laptev et al. [23] (2008)	91.8%
Niebles et al. [27] (2010)	91.3%
Castrodad et al. [8] (2012)	96.3%
Alfaro et al. [3] (2013)	95.7%
Our method	**97.5%**

Table 2. Recognition rates of our and alternative methods on KTH dataset. In all cases, the same testing protocol is used.

Olympic Dataset: This dataset contains 16 actions corresponding to 783 videos of athletes practicing different sports [27]. Fig. 5 shows sample frames displaying the action classes. In our experiments, we use the original setup [27] to divide the data into training and test sets. Table 3 shows the recognition performance reached by our method and several alternative state-of-the-art techniques. Our approach achieves a recognition rate of 96.3%. This is a remarkable increase in performance with respect to previous state-of-the-art approaches. Fig. 5 shows the confusion matrix reported by our method. We note that many actions from this dataset have a perfect recognition rate. Therefore, our approach effectively captures relevant acts and their temporal relationships.

Method	Acc.
Niebles et al. [27] (2010)	72.1%
Liu et al. [25] (2011)	74.4%
Jiang et al. [18] (2012)	80.6%
Alfaro et al. [3] (2013)	81.3%
Gaidon et al. [15] (2014)	85.0 %
Our method	**96.3%**

Table 3. Recognition rates of our and alternative methods on Olympic dataset. In all cases, the same testing protocol is used.

Hollywood Dataset: This dataset contains video clips extracted from 32 movies and displaying 8 action classes. Fig. 6 shows sample frames displaying the action classes. We use only the videos with manual annotations (clean training file) and we limit the dataset to videos with a single label. This is the same testing protocol used by the alternative techniques considered here. Table 4 shows the recognition performance of our method and several alternative state-of-the-art techniques. Our approach achieves a recognition rate of 71.9%. Again, this is a remarkable increase in performance with respect to previous state-of-the-art approaches. Fig. 6 shows the confusion matrix reported by our method. Actions such as *answer phone*, *handshake*, and *hug person* obtain high recognition rates. In contrast, the actions *get out car*, *kiss*, and *sit up* present a lower recognition performance. According to the confusion matrix in Fig. 6, these actions present a high confusion rate with respect to the action *answer phone*. This can be explained by the presence of a common pattern among these actions in this dataset, which is given by a slow incorporation of the main actor.

Method	Acc.
Laptev et al. [23] (2008)	38.4%
Wang et al. [38] (2009)	47.4%
Wu et al. [41] (2011)	47.6%
Zhou et al. [46] (2015)	50.5%
Our method	**71.9%**

Table 4. Recognition rates of our and alternative methods on HOHA dataset. In all cases, the same testing protocol is used.

4.2. Evaluation of Method to Extract Key-Sequences

In this section, we evaluate the relevance of the proposed method to obtain key-frames by replacing this step of our approach by alternative strategies. Besides this modification, we maintain the remaining steps of our approach and use the same parameter values reported in Section 4.1. In particular, we implement two baselines, Table 5 shows our results:
Baseline 1 (B1), *Uniform Selection*: We split the video into K equal-sized temporal segments and select the central frame of each segment as a key-frame.
Baseline 2 (B2), *K-Means*: We generate all possible video sequences containing $2t + 1$ frames by applying a temporal sliding window. We then apply the K-Means clustering algorithm to each class to obtain K cluster centers per class. For each video, we select as key-frames the most similar descriptor to each cluster center.

4.3. Evaluation of ITRA descriptor

We evaluate the effectiveness of our ITRA descriptor by replacing this part of our approach by alternative schemes to obtain a video descriptor. These alternative schemes are

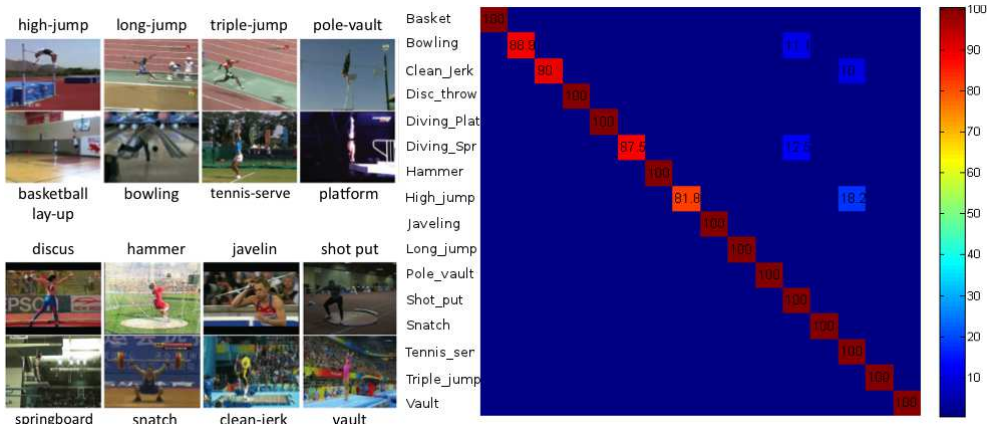

Figure 5. Olympic dataset. Left: sample from each action class. Right: confusion matrix for our method on Olympic dataset.

Figure 6. HOHA. Left: sample from each action class. Right: confusion matrix for our method on HOHA dataset.

Dataset	Method		
	B1	B2	Ours
HOHA	34.2%	37.2%	**71.9%**
Olympic	46.3%	63.4%	**96.3%**

Table 5. Performances of our method and alternative strategies to extract key-sequences.

Dataset	Method		
	B1	B2	Ours
HOHA	42.2%	51.3%	**71.9%**
Olympic	72.4%	87.3%	**96.3%**

Table 6. Performances of our method and alternative strategies to construct the video descriptor using sparse coding techniques.

also based on sparse coding techniques but they do not exploit relative local or temporal information. Specifically, we consider two baselines, Table 6 shows our results:

Baseline 1 (B1), *Ignoring relative local temporal information*: All key-sequences from all temporal positions are combined to build a single class-shared joint dictionary that do not preserve temporal order among the key-sequences. This baseline can be considered as a BoW type of representation that does not encode relative temporal relations among key-sequences.

Baseline 2 (B2), *Ignoring intra-class relations*: this baseline only considers the term in ITRA descriptor associated to the *Inter-Class Relative Act Descriptor* $\mathbf{\Phi}^i$, discarding intra-class relations provided by the *Intra-Class Relative Act Descriptor* $\mathbf{\Psi}^i$.

5. Conclusions

We present a novel method for category-based action recognition in video. As a main result, our experiments show that the proposed method reaches remarkable action recognition performance on 3 popular benchmark datasets. Furthermore, the reduced dimensionality of the ITRA descriptor provides a fast classification scheme. Actually, using a reduced dimensionality, between 24 and 54 dimensions for the datasets considered here, it provides a representation that demonstrates to be highly discriminative. As future work, the ITRA descriptor opens the possibility to explore several strategies to concatenate the basic dictionaries to access different relative similarity relationships.

Acknowledgements

This work was partially funded by FONDECYT grant 1151018, CONICYT, Chile.

References

[1] J. Aggarwal and M. Ryoo. Human activity analysis. *ACM Computing Surveys*, 43(3):1–43, 2011. 2

[2] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. on Signal Processing*, 54:4311–4322, 2006. 5

[3] A. Alfaro, D. Mery, and A. Soto. Human action recognition from inter-temporal dictionaries of key-sequences. In *Pacific-Rim Symposium on Image and Video Technology*, 2013. 7

[4] J. Assa, Y. Caspi, and D. Cohen-Or. Action synopsis: Pose selection and illustration. In *SIGGRAPH*, 2005. 1

[5] I. Atmosukarto, B. Ghanem, and N. Ahuja. Trajectory-based fisher kernel representation for action recognition in videos. In *Int. Conf. on Pattern Recognition*, 2012. 2

[6] A. Bosch, A. Zisserman, and X. Muñoz. Representing shape with a spatial pyramid kernel. In *International Conference on Image and Video Retrieval*, pages 401–408, 2007. 3

[7] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *European Conf. on Computer Vision*, 2010. 2

[8] A. Castrodad and G. Sapiro. Sparse modeling of human actions from motion imagery. *Int. Journal of Computer Vision*, 100:1–15, 2012. 3, 7

[9] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005. 2

[10] D. Donoho and M. Elad. Optimally sparse representation in general (non orthogonal) dictionaries via l_1 minimization. In *Proc. of the National Academy of Sciences*, 2003. 3

[11] E. Elhamifar, G. Sapiro, and R. Vidal. See all by looking at a few: Sparse modeling for finding representative objects. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012. 3

[12] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645. 2

[13] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite-element approximations. *Computers & Mathematics with Applications*, 2:17–40, 1976. 3

[14] A. Gaidon, Z. Harchaoui, and C. Schmid. Actom sequence models for efficient action detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2011. 1, 2

[15] A. Gaidon, Z. Harchaoui, and C. Schmid. Activity representation with motion hierarchies. *IJCV*, 107(3):219–238, 2014. 7

[16] T. Guha and R. Ward. Learning sparse representations for human action recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34:1576–1588, 2012. 3, 6

[17] K. Guo, P. Ishwar, and J. Konrad. Action recognition in video by sparse representation on covariance manifolds of silhouette tunnels. In *Int. Conf. on Pattern Recognition*, 2010. 3

[18] Y. Jiang, Q. Dai, X. Xue, W. Liu, and C. Ngo. Trajectory-based modeling of human actions with motion reference points. In *European Conf. on Computer Vision*, 2012. 7

[19] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *IEEE Int. Conf. on Computer Vision*, 2005. 1

[20] A. Kläser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *British Machine Vision Conference*, 2008. 2, 4

[21] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Attribute and simile classifiers for face verification. In *IEEE Int. Conf. on Computer Vision*, 2009. 2

[22] I. Laptev and T. Lindeberg. Space-time interest points. In *IEEE Int. Conf. on Computer Vision*, 2003. 2

[23] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008. 2, 6, 7

[24] J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008. 2

[25] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2011. 2, 7

[26] L. Liu, L. Shao, and P. Rockett. Boosted key-frame selection and correlated pyramidal feature representation for human action recognition. *Pattern Recognition*, 46:1810–1818, 2013. 2

[27] J. Niebles, C. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *European Conf. on Computer Vision*, 2010. 1, 2, 6, 7

[28] D. Oneata, J. Verbeek, and C. Schmid. Action and event recognition with fisher vectors on a compact feature set. In *IEEE Int. Conf. on Computer Vision*, 2013. 2

[29] D. Parikh and K. Grauman. Relative attributes. In *IEEE Int. Conf. on Computer Vision*, 2011. 2

[30] M. Raptis, I. Kokkinos, and S. Soatto. Discovering discriminative action parts from mid-level video representations. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012. 1

[31] M. Raptis and L. Sigal. Poselet key-framing: A model for human activity recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2013. 2

[32] S. Sadanand and J. Corso. Action Bank: A high-level representation of activity in video. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012. 2

[33] K. Schindler and L. V. Gool. Action Snippets: How many frames does human action recognition require? In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008. 1, 2

[34] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *Int. Conf. on Pattern Recognition*, 2004. 6, 7

[35] K. Tran, I. Kakadiaris, and S. Shah. Modeling motion of body parts for action recognition. In *BMVC*, 2011. 3

[36] G. Wang, D. A. Forsyth, and D. Hoiem. Improved object categorization and detection using comparative object similarity. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(10):2442–2453, 2013. 2

[37] H. Wang, A. Kläser, C. Schmid, and C. Liu. Dense trajectories and motion boundary descriptors for action recognition. *Int. Journal of Computer Vision*, 103(1):60–79, 2013. 2

[38] H. Wang, M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference*, 2009. 7

[39] Y. Wang and G. Mori. Hidden part models for human action recognition: Probabilistic versus max margin. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(7):1310–1323, 2011. 2

[40] D. Weinland and E. Boyer. Action recognition using exemplar-based embedding. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008. 1, 2

[41] S. Wu, O. Oreifej, and M. Shah. Action recognition in videos acquired by a moving camera using motion decomposition of lagragian particle trajectories. In *IEEE Int. Conf. on Computer Vision*, 2011. 7

[42] J. Yagnik, D. Strelow, D. Ross, and R. Lin. The power of comparative reasoning. In *IEEE Int. Conf. on Computer Vision*, 2011. 1, 2

[43] A. Yao, U. Gall, and L. V. Gool. A Hough transform-based voting framework for action recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2010. 3

[44] T. Yu, T. Kim, and R. Cipolla. Real-time action recognition by spatiotemporal semantic and structural forest. In *British Machine Vision Conference*, 2010. 2, 3

[45] Z. Zhao and A. Elgammal. Information theoretic key frame selection for action recognition. In *British Machine Vision Conference*, 2008. 1, 2

[46] Z. Zhou, F. Shi, and W. Wu. Learning spatial and temporal extents of human actions for action detection. *IEEE Transactions on multimedia*, 17(4):512–525, 2015. 7

[47] X. Zhu, D. Anguelov, and D. Ramanan. Capturing longtail distributions of object subcategories. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2014. 1