Int J Comput Vis (2012) 100:1–15
DOI 10.1007/s11263-012-0534-7

Sparse Modeling of Human Actions from Motion Imagery

Alexey Castrodad · Guillermo Sapiro

Received: 31 July 2011 / Accepted: 27 April 2012 / Published online: 6 June 2012

Abstract An efficient sparse modeling pipeline for the classification of human actions from video is here developed. Spatio-temporal features that characterize local changes in the image are first extracted. This is followed by the learning of a class-structured dictionary encoding the individual actions of interest. Classification is then based on reconstruction, where the label assigned to each video comes from the optimal sparse linear combination of the learned basis vectors (action primitives) representing the actions. A low computational cost deep-layer model learning the inter-class correlations of the data is added for increasing discriminative power. In spite of its simplicity and low computational cost, the method outperforms previously reported results for virtually all standard datasets.

Keywords Action classification · Sparse modeling · Dictionary learning · Supervised learning

1 Introduction

We are living in an era where the ratio of data acquisition over exploitation capabilities has dramatically exploded. With this comes an essential need for automatic and semi-automatic tools that could aid with the processing requirements in most technology-oriented fields. A clear example pertains to the surveillance field, where video feeds from possibly thousands of cameras need to be analyzed by a limited amount of operators on a given time lapse. As simple as it seems for us to recognize human actions, it is still not well understood how the processes in our visual system give our ability to interpret these actions, and consequently is difficult to effectively emulate these through computational approaches. In addition to the intrinsic large variability for the same type of actions, factors like noise, camera motion and jitter, highly dynamic backgrounds, and scale variations, increase the complexity of the scene, therefore having a negative impact in the performance of the classification system. In this paper, we focus in a practical design of such a system, that is, an algorithm for supervised classification of human actions in motion imagery.

There are a number of important aspects of human actions and motion imagery in general that make the particular task of action classification very challenging:

1. Data is very high dimensional and redundant: Each video will be subdivided into spatio-temporal patches which are then vectorized, yielding high-dimensional data samples. Redundancy occurs from the high temporal sampling rate, allowing relatively smooth frame-to-frame transitions, hence the ability to observe the same object many times (not considering shot boundaries). In addition, many (but not all) of the actions have an associated periodicity of movements. Even if there is no periodicity associated with the movements, the availability of training data implies that the action of interest will be observed redundantly, since overlapping patches characterizing a specific spatio-temporal behavior are generally very similar, and will be accounted multiple times with relatively low variation. These properties of the data allow the model to benefit from the *blessings* of high dimensionality (Donoho 2000), and will be key to over-

Alexey Castrodad is also with NGA.

A. Castrodad (✉) · G. Sapiro
Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA
e-mail: castr103@umn.edu

G. Sapiro
e-mail: guille@umn.edu

coming noise and jitter effects, allowing simple data representations by using simple features, while yielding stable and highly accurate classification rates.

2. Human activities are very diverse: Two people juggling a soccer ball can do that very differently. Same for people swimming, jumping, boxing, or performing any of the activities we want to classify. Learning simple representations is critical to address such variability.
3. Different human activities share common movements: A clear example of this is the problem of distinguishing if a person is either running or jogging. Torso and arms movements may be very similar for both actions. Therefore, there are spatio-temporal structures that are shared between actions. While one would think that a person running moves faster than a person jogging, in reality it could be the exact opposite (consider racewalking). This phenomena suggests that our natural ability to classify actions is not based only on local observations (e.g., torso and arms movements) or global observations (e.g., person's velocity) but on local *and* global observations. This is consistent with recent psychological research indicating that the perception of human actions are a combination of spatial hierarchies of the human body along with motion regularities (Blake and Shiffrar 2007). Relationships between activities play an important role in order to compare among them, and this will be incorporated in our proposed framework via a simple deep learning structure.
4. Variability in the video data: While important applications, here addressed as well, consist of a single acquisition protocol, e.g., surveillance video; the action data we want to classify is often recorded in a large variety of scenarios, leading to different viewing angles, resolution, and general quality. This is the case for example of the YouTube data we will use as one of the testing scenarios for our proposed framework.

In this paper, we consider these aspects of motion imagery and human actions and propose a hierarchical, two-level sparse modeling framework that exploits the high dimensionality and redundancy of the data. Differently from the recent literature, discussed in Sect. 2, we learn inter-class relationships using both global and local perspectives. As described in detail in Sect. 3, we combine ℓ_1-minimization with structured dictionary learning, and show that with proper modeling, in combination with a reconstruction and complexity based classification procedure using sparse representations, a *single feature* and a *single sampling scale* are sufficient for highly accurate activity classification on a large variety of examples.

We claim that there is a great deal of information inherent in the sparse representations that have not yet been fully explored. In Mairal et al. (2008) for example, class-decision functions were incorporated in the sparse modeling optimization to gain higher discriminative power. In the results the authors show that significant gain can be attained for recognition tasks, but always at the cost of more sophisticated modeling and optimizations. We drift away from these ideas by explicitly exploiting the sparse coefficients in a different way such that, even though it derives from a purely generative model, takes more advantage from the structure given in the dictionary to further model class distributions with a simpler model. In Sect. 4 we evaluate the performance of the model using four publicly available datasets: the KTH Human Action Dataset, the UT-Tower Dataset, the UCF-Sports Dataset, and the YouTube Action Dataset, each posing different challenges and environmental settings, and compare our results to those reported in the literature. Our proposed framework uniformly produces state-of-the-art results for all these data, exploiting a much simpler modeling than those previously proposed in the literature. Finally, we provide concluding remarks and future research in Sect. 5.

2 Related Work

The recently proposed schemes for action classification in motion imagery are mostly feature-based. These techniques include three main steps. The first step deals with "interest point detection," and it consists of searching for spatial and temporal locations that are appropriate for performing feature extraction. Examples are Cuboids (Dollar et al. 2005), Harris3D (Laptev and Lindeberg 2003), Hessian (Willems et al. 2008), and dense sampling[1] (Gall et al. 2011; Le et al. 2011; Wang et al. 2011). This is followed by a "feature acquisition" step, where the video data at the locations specified from the first step undergo a series of transformation processes to obtain descriptive features of the particular action, many of which are derived from standard static scene and object recognition techniques. Examples are SIFT (Scovanner et al. 2007), the Cuboids feature (Dollar et al. 2005), Histograms of Oriented Gradients (HOGs) (Laptev et al. 2008), and its extension to the temporal domain, i.e., HOG3D (Kläser et al. 2008), combinations of HOG and Histograms of Optical Flow (HOF) (Laptev et al. 2008), Extended Speeded Up Robust Features (ESURF), Local Trinary Patterns (Yeffet and Wolf 2009), and Motion Boundary Histograms (MBH) (Dalal and Triggs 2006). Finally, the third step is a "classification/labeling" process, where bag-of-features consisting of the features extracted (or vector quantized versions) from the second step are fed into a classifier, often a Support Vector Machine (SVM). Please

[1] Dense sampling is not an interest point detector *per se*. It extracts spatio-temporal multi-scale patches indiscriminately throughout the video at all locations.

refer to Shao and Mattivi (2010) and Wang et al. (2009) for comprehensive reviews and pointers to feature-based as well as other proposed schemes.

In practice, it is difficult to measure what combinations of detectors and features are best for modeling human actions. In Wang et al. (2009), the authors conducted exhaustive comparisons on the classification performance of several spatio-temporal interest point detectors and descriptors using nonlinear SVMs, using publicly available datasets. They observed that most of the studied features performed relatively well, although their individual performance was very dependent on the dataset. For example, interest point detection based feature extraction performed better than dense sampling on datasets with relatively low complexity like KTH, while dense sampling performed slightly better in more realistic/challenging datasets like UCF-Sports. In this work, we do not look at designing detectors or descriptors but rather give greater attention into developing a powerful model for classification using sparse modeling. We use a very simple detector and descriptor, and one single spatio-temporal scale to better show that sparse modeling is capable of taking high dimensional and redundant data and translate it into highly discriminative information. Also, given that the gain in performance of dense sampling is not significant, and it takes longer computation times, we use a simple interest point detector (by thresholding) instead of dense sampling, simply for a faster and more efficient sampling process, such that the spatio-temporal patches selected contain slightly higher velocity values relative to a larger background.

Sparse coding along with dictionary learning has proven to be very successful in many signal and image processing tasks, especially after highly efficient optimization methods and supporting theoretical results emerged. More recently, it has been adapted to classification tasks like face recognition (Wright et al. 2008) (without dictionary learning), digit and texture classification (Mairal et al. 2008; Ramirez et al. 2010), hyperspectral imaging (Castrodad et al. 2011; Charles et al. 2011), among numerous other applications. It has also been applied recently for motion imagery analysis for example in Cadieu and Olshausen (2008), Dean et al. (2009), Guo et al. (2010), Taylor et al. (2010). In Dean et al. (2009), the authors propose to learn a dictionary in a recursive manner by first extracting high response values coming from the Cuboids detector, and then using the resulting sparse codes as the descriptors (features), where PCA is optionally applied. Then, as often done for classification, the method uses a bag-of-features with K-bin histograms approach for representing the videos. To classify unlabeled videos, these histograms are fed into a nonlinear χ^2-SVM. In contrast to our work, the authors learn a basis globally, while the proposed method learns it in a per-class manner, and follows a different scheme for classification. We also learn inter-class relationships via a two levels (deep-learning) approach.

In Guo et al. (2010), the authors build a dictionary using vectorized log-covariance matrices of 12 hand-crafted features (mostly derived from optical flow) obtained from entire labeled videos. Then, the vectorized log-covariance matrix coming from an unlabeled video is represented with this dictionary using ℓ_1-minimization, and the video is classified by selecting the label associated with those dictionary atoms that yield minimum reconstruction error. In contrast to our work, the dictionary in Guo et al. (2010) is hand-crafted directly from the training data and not learned. While similar in nature to the ℓ_1-minimization procedure used in our first level, the data samples in Guo et al. (2010) are global representations of the entire video, while our method first models all local data samples (spatio-temporal patches), followed by a fast global representation on a second stage, leading to a hierarchical model that learns both efficient per-class representations (first level) as well as inter-class relationships (second level).

In Jhuang et al. (2007), the authors propose a three-level algorithm that simulates processes in the human visual cortex. These three levels use feature extraction, template matching, and max-pooling to achieve both spatial and temporal invariance by increasing the scale at each level. Classification of these features is performed using a sparsity inducing SVM. Compared to our model, except for the last part of its second level, the features are hand-crafted, and is overall a more sophisticated methodology.

In Taylor et al. (2010), a convolutional Restricted Boltzmann Machine (convRBM) architecture is applied to the video data for learning spatio-temporal features by estimating frame-to-frame transformations implicitly. They combine a series of sparse coding, dictionary learning, and probabilistic spatial and temporal pooling techniques (also to yield spatio-temporal invariance), and then feed sparse codes that are max-pooled in the temporal domain (emerging from the sparse coding stage) into an RBF-SVM. Compared to our work, this method deals with expensive computations on a frame by frame basis, making the training process very time consuming. Also they train a global dictionary of all actions. In contrast, our method learns per-class/activity dictionaries independently using corresponding training data all at once (this is also beneficial when new classes appear, no need to re-train the entire dictionary). In Le et al. (2011), Independent Subspace Analysis (ISA) networks are applied for learning from the data using two levels. Blocks of video data are used as input to the first ISA network following convolution and stacking techniques. Then, to achieve spatial invariance, the combined outputs from the first level are convolved with a larger image area and reduced in size using PCA, and then fed to the second level, another ISA network. The outputs from this level

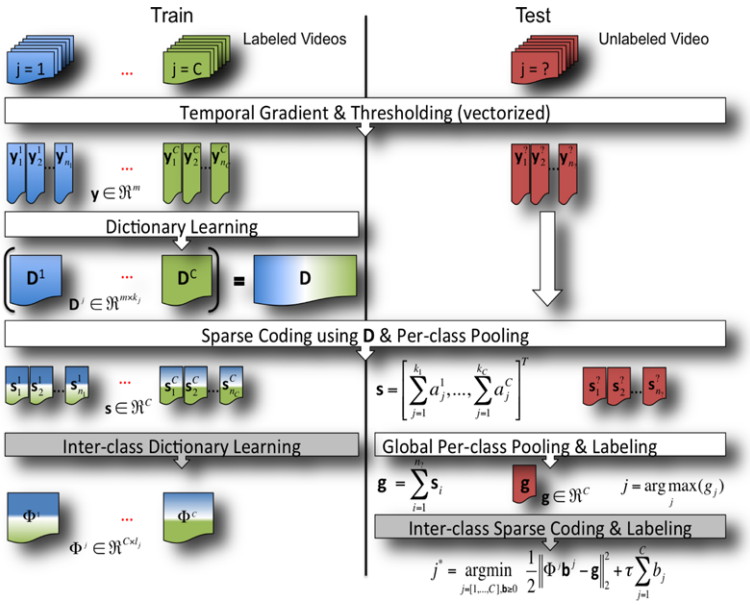

Fig. 1 Algorithm overview. The *left* and *right sides* illustrate the learning and classification procedures, respectively. The processes in *white boxes* represent the first level of sparse modeling. The processes in *gray boxes* represent the second level (Color figure online)

are vector quantized (bag-of-features approach), and a χ^2-SVM is used for classification. The method here proposed does not uses PCA to reduce the dimensionality of the data after the first level, as the dimension reduction derives more directly and naturally by using sum-pooling in a per-class manner after the first level.

Note that the hierarchical modeling of the proposed method is different from Jhuang et al. (2007), Le et al. (2011), and Taylor et al. (2010). These works progress from level to level by sequentially increasing spatial and/or temporal scales, thus benefiting from a multi-scale approach (spatial invariance), while our work progresses from locally oriented representations using only one scale,[2] to a globally oriented video representation deriving directly from the sparse model, and not from a bag-of-features approach or series of multi-scale pooling mechanisms. Also, the proposed scheme, as we will discuss in more detail next, produces sparse codes that contain information in a different way than the sparse codes produced with the global dictionaries in Dean et al. (2009), Taylor et al. (2010). This is achieved by explicit per-class learning and pooling, yielding a C-space, for C activities, representation with invariance to the per-class selection of action primitives (learned basis).

3 Sparse Modeling for Action Classification

3.1 Model Overview

Assume we have a set of labeled videos, each containing 1 of C known actions (classes) with associated label $j \in [1, 2, \ldots, C]$.[3] Our goal is to learn from these labeled videos in order to classify new incoming unlabeled ones, and achieve this via simple and computationally efficient paradigms. We solve this with a two-level feature-based scheme for supervised learning and classification, which follows the pipeline shown in Fig. 1.

For learning, we begin with a set of labeled videos, and for each action separately, we extract and vectorize overlapping spatio-temporal patches consisting of the videos' temporal gradients at locations that are above a pre-defined energy threshold. In other words, we exploit spatio-temporal ($3D$) patches that have sufficient activity. During the *first level of training*, these labeled training samples (i.e., $\mathbf{y}^j$ vectors from patches belonging to videos of class j) serve as input to a dictionary learning stage. In this stage, an action-specific dictionary $\mathbf{D}^j$ of k_j atoms is learned for each of the C classes. After learning all C dictionaries, a structured dictionary $\mathbf{D}$ consisting of the concatenation of these sub-dictionaries is formed. A sparse representation of these training samples (spatio-temporal $3D$ patches) using ℓ_1-minimization yields associated sparse coefficients vectors.

[2] In this work, only a single scale is used to better illustrate the model's advantages, already achieving state-of-the-art results. A multi-scale approach could certainly be beneficial.

[3] In this work, as commonly done in the literature, we assume each video has been already segmented into time segments of uniform (single) actions. Considering we will learn and detect actions based on just a handful of frames, this is not a very restrictive assumption. We will comment more on this later in the paper.

These coefficient vectors are pooled in a per-class manner, so that they quantify the contribution from each action (i.e., the $\mathbf{s}^j$ vectors, each patch of class j producing one). Then, on a *second level of training*, these per-class pooled samples become the data used for learning a second set of action-specific dictionaries $\mathbf{\Phi}^j$ of l_j atoms. While the first level dictionaries $\mathbf{D}^j$ are class independent, these second level ones model the inter-relations between the classes/actions. With this, the off-line learning stage of the algorithm concludes.

To classify a video with unknown label "?," we follow the same feature extraction procedure, where test samples, $\mathbf{y}^?$'s (again consisting of spatio-temporal patches of the video's temporal gradient) are extracted and sparsely represented using the (already learned) structured dictionary $\mathbf{D}$. After sparse coding, the resulting vectors of coefficients are also pooled in a per-class manner, yielding the $\mathbf{s}^?$'s vectors. For a sometimes sufficient first level classification, a label is assigned to the video by majority voting, that is, the class with the largest contribution using all the pooled vectors is selected. For a second level classification, the same majority voted single vector is sparsely represented using the concatenation of all the dictionaries $\mathbf{\Phi}^j$. The video's label j^* is selected such that the representation obtained with the jth action subdictionary $\mathbf{\Phi}^j$ yields the minimum sparsity *and* reconstruction trade-off.

We now give a detailed description of the proposed modeling and classification algorithm for activity classification. We start with the data representation and feature extraction process, which is the same for labeled (training) and unlabeled (testing) videos. Then, we describe the first level of sparse modeling, where dictionaries are learned for each of the actions. This is followed by the second level of the learning process, where a new set of dictionaries are learned to model inter-class relationships. We finalize this section with a description of the labeling/classification procedure, and briefly contrast the model here proposed with the bag of features approach.

3.2 Data Representation and Modeling

Let $\mathbf{I}$ be a video, and $\mathbf{I}_t$ its temporal gradient. In order to extract informative spatio-temporal patches, we use a simple thresholding operation. More precisely, let $\mathbf{I}_t(p)$ be a $3D$ (space+time) patch of I_t with center at location $p \in \Omega$, where Ω is the video's spatial domain. Then, we extract data samples $\mathbf{y}(p) = \text{vect}(|\mathbf{I}_t(p)|)$ such that $|I_t(p)| > \delta, \forall p$, where δ is a pre-defined threshold, and vect$(\cdot)$ denotes vectorization (in other words, we consider spatio-temporal patches with above threshold temporal activity). Let all the data extracted from the videos this way be denoted by $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n] \in \Re^{m \times n}$, where each column $\mathbf{y}$ is a data sample. Here m is then the data dimension $m = r \times c \times w$, where r, c, and w are the pre-defined number of rows, columns, and frames of the spatio-temporal patch, respectively, and n the number of extracted "high-activity" patches.

We model the data samples linearly as $\mathbf{y} = \mathbf{D}\mathbf{a} + \mathbf{n}$, where $\mathbf{n}$ is an additive component with bounded energy ($\|\mathbf{n}\|_2^2 \leq \epsilon$) modeling both the noise and the deviation from the model, $\mathbf{a} \in \Re^k$ are the approximation weights, and $\mathbf{D} \in \Re^{m \times k}$ is a (possibly overcomplete, $k > m$) to be learned dictionary. Assuming for the moment that $\mathbf{D}$ is fixed, a sparse representation of a sample $\mathbf{y}$ is obtained as the solution to the following optimization problem:

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} \|\mathbf{a}\|_0 \quad \text{s.t. } \frac{1}{2}\|\mathbf{D}\mathbf{a} - \mathbf{y}\|_2^2 \leq \epsilon, \tag{1}$$

where $\|\cdot\|_0$ is a pseudo-norm that counts the number of nonzero entries. This means that the spatio-temporal patches belong to the low dimensional subspaces defined by the dictionary $\mathbf{D}$. Under assumptions on the sparsity of the signal and the structure of the dictionary $\mathbf{D}$ (see Bruckstein et al. 2009), there exists $\lambda > 0$ such that (1) is equivalent to solving

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} \frac{1}{2}\|\mathbf{D}\mathbf{a} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{a}\|_1, \tag{2}$$

known as the Lasso (Tibshirani 1994). Notice that the ℓ_0 pseudo norm was replaced by an ℓ_1-norm, and we prefer in our work the formulation in (2) over the one in (1) since it is more stable and easily solvable using modern convex optimization techniques.

The dictionary $\mathbf{D}$ can be constructed for example using wavelets basis. However, in this work, since we know instances of the signal, we learn/infer the dictionary using training data, bringing the advantage of a better data fit compared with the use of off-the-shelf dictionaries. Contrasting with sparse coding, we denote this process of also learning the dictionary *sparse modeling*. Sparse modeling of data can be done via an alternation minimization scheme similar in nature to K-means, where we fix $\mathbf{D}$, obtain the sparse code $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n] \in \Re^{k \times n}$, then minimizing with respect to $\mathbf{D}$ while fixing $\mathbf{A}$ (both sub-problems are convex), and continue this process until reaching a (local) minimum to get

$$\left(\mathbf{D}^*, \mathbf{A}^*\right) = \arg\min_{\mathbf{D},\mathbf{A}} \frac{1}{2}\|\mathbf{D}\mathbf{A} - \mathbf{Y}\|_F^2 + \lambda \sum_{i=1}^{n} \|\mathbf{a}_i\|_1, \tag{3}$$

which can be efficiently solved using algorithms like the K-SVD (Aharon et al. 2006; Mairal et al. 2010).

This concludes the general formulation for feature extraction and data representation using sparse modeling. Next, we focus our attention on a supervised classification setting, specifically applied to action classification.

3.2.1 Learning Action-Specific Dictionaries

Since we are in the supervised setting, there are labeled training data available for each of the actions. Let $\mathbf{Y}^j =$

$[\mathbf{y}_1^j, \ldots, \mathbf{y}_{n_j}^j] \in \Re^{m \times n_j}$ be the n_j extracted samples corresponding to the jth action/class. We obtain the jth action representation (class-specific dictionary) $\mathbf{D}^j \in \Re_+^{m \times k_j}$ by solving

$$\mathbf{D}^{j*} = \arg \min_{(\mathbf{D}^j, \mathbf{A}^j) \succeq 0} \frac{1}{2} \left\| \mathbf{D}^j \mathbf{A}^j - \mathbf{Y}^j \right\|_F^2 + \lambda \sum_{i=1}^{n_j} \mathcal{S}\left(\mathbf{a}^j\right), \quad (4)$$

where $(\mathbf{a} \succeq \mathbf{b})$ denotes the element-wise inequality, and $\mathcal{S}(\mathbf{a}^j) = \sum_{i=1}^{k_j} a_i^j$. Notice that we modified the sparse modeling formulation of (3) to a nonnegative version, and this can be interpreted as performing a sparsity constrained nonnegative matrix factorization on each class. We repeat this procedure and learn dictionaries for all C classes. As we explain next, these compose the overall actions structured dictionary D.

3.2.2 *Modeling Local Observations as Mixture of Actions: Level-1*

Once the action-dependent dictionaries are learned, we express each of the data samples (extracted spatio-temporal patches with significant energy) as sparse linear combinations of the different actions by forming the block-structured dictionary $\mathbf{D} = [\mathbf{D}^1, \ldots, \mathbf{D}^C] \in \Re_+^{m \times k}$, where $k = \sum_{j=1}^C k_j$. Then we get, for the entire data being processed Y,

$$\mathbf{A}^* = \arg \min_{\mathbf{A} \succeq 0} \frac{1}{2} \|\mathbf{D}\mathbf{A} - \mathbf{Y}\|_F^2 + \lambda \sum_{i=1}^{n} \mathcal{S}(\mathbf{a}_i), \quad (5)$$

where $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n] \in \Re_+^{k \times n}$, $\mathbf{a}_i = [a_i^1, \ldots, a_i^{k_1}, \ldots, a_i^{k_C}]^T \in \Re_+^k$, and $n = \sum_{j=1}^C n_j$. Note that this includes all the high energy spatio-temporal patches from all the available training videos for all the classes.

Note that with this coding strategy, we are expressing the data points (patches) as a sparse linear combination of elements of the entire structured dictionary $\mathbf{D}$, not only of their corresponding class-dependent subdictionary (see also Wright et al. (2008) for a related coding strategy for facial recognition). That is, each data sample becomes a "mixture" of the actions modeled in $\mathbf{D}$, and the component (or fraction) of the jth action mixture is given by its associated $\mathbf{a}^j$. The idea is to quantify movement sharing between actions. If none of the local movements associated with the jth action are shared, then the contribution from the other action representations will be zero, meaning that the data sample is purely pertaining of the jth action, and is quantified in $\mathcal{S}(\mathbf{a}^j)$. On the other hand, shared movements will be quantified with nonzero contributions from more than one class, meaning that the data samples representing these may lie in the space of other actions. This strategy permits to share features between actions, and to represent actions not only by their own model but also by how connected they are to the models of other actions. This cross-talking between the different action's models (classes) will be critical in the second stage of the learning model, as will be detailed below. The sparsity induced in the minimization should reduce the number of errors caused by this sharing effect. Furthermore, these mixtures can be modeled by letting $\mathbf{s} = [\mathcal{S}(\mathbf{a}^1), \ldots, \mathcal{S}(\mathbf{a}^C)]^T \in \Re_+^C$ be the per-class ℓ_1-norm vector corresponding to the data sample $\mathbf{y}$, and letting $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_n] \in \Re_+^{C \times n}$ be the matrix of all per-class ℓ_1-norm samples. By doing this, the actions' contributions in the sample are quantified with invariance to the subset selection in the sub-dictionaries $\mathbf{D}^j$, and the dimensionality of the data is notably reduced to C-dimensional vectors in a reasonable way, as opposed to an arbitrary reduction using for example PCA. This reduced dimension, which again expresses the inter-class (inter-action) components of the data, low dimensional input to the next level of the learning process.

3.2.3 *Modeling Global Observations: Level-2*

Once we obtain the characterization of the data in terms of a linear mixture of the C actions, we begin our second level of modeling. Using the training data from each class, $\mathbf{S}^j \in \Re_+^{C \times n_j}$ (the C-dimensional s^j vectors for class j), we model inter-class relationships by learning a second set of per-class dictionaries $\mathbf{\Phi}^j \in \Re_+^{C \times l_j}$ as:

$$\mathbf{\Phi}^{j*} = \arg \min_{(\mathbf{\Phi}^j, \mathbf{B}^j) \succeq 0} \frac{1}{2} \left\| \mathbf{\Phi}^j \mathbf{B}^j - \mathbf{S}^j \right\|_F^2 + \tau \sum_{i=1}^{n_j} \mathcal{S}\left(\mathbf{b}^j\right), \quad (6)$$

where $\mathbf{B}^j = [\mathbf{b}_1^j, \ldots, \mathbf{b}_{n_j}^j] \in \Re^{l_j \times n_j}$ are the associated sparse coefficients from the samples in the $j-th$ class, and $\tau > 0$ controls the trade-off between class reconstruction and coefficients' sparsity. Notice that although the dictionaries $\mathbf{\Phi}^j$ are learned on a per-class basis, each models how data samples corresponding to a particular action j can have energy contributions from other actions, since they are learned from the n_j mixed coefficients $\mathbf{s}^j \in \Re_+^C$. Inter-class (actions) relationships are then learned this way.

This completes the description of the modeling as well as the learning stage of the proposed framework. We now proceed to describe how is this modeling exploited for classification.

3.3 Classification

In the first level of our hierarchical algorithm, we learned dictionaries using extracted spatio-temporal samples from the labeled videos. Then, each of these samples are expressed as a linear combination of all the action dictionaries

to quantify the amount of action mixtures. After class sum-pooling (ℓ_1-norm on a per-class basis) of the corresponding sparse coefficients, we learned a second set of dictionaries modeling the overall per-class contribution per sample. We now describe two decision rules for classification that derive directly from each modeling level.

3.3.1 Labeling After Level 1

It is expected that the information provided in $\mathbf{S}$ should be already significant for class separation. Let $\mathbf{g} = \mathbf{S1} \in \Re_+^C$, where $\mathbf{1}$ is a $n \times 1$ vector with all elements one (note that now n is the amount of spatio-temporal patches with significant energy present in a *single* video being classified). Then, we classify a video according to the mapping function $f_1(\mathbf{g}) : \Re_+^C \rightarrow \mathcal{Z}$ defined as

$$f_1(\mathbf{g}) = \big\{ j | g_j > g_i, j \neq i, (i, j) \in [1, \ldots, C] \big\}. \tag{7}$$

This classification, already provides competitive results, especially with actions that do not share too many spatio-temporal structures, see Sect. 4. The second layer, that due to the significant further reduction in dimensionality (to C, the number of classes), is computationally negligible, improves the classification even further.

3.3.2 Labeling After Level 2

There are cases where there are known shared (local) movements between actions, or cases where a video is composed of more than one action (e.g., running and then kicking a ball). As discussed before, the first layer is not yet exploiting inter-relations between the actions. Inspired in part on ideas from Sprechmann and Sapiro (2010), we develop a classification scheme for the second level. Let

$$\mathcal{R}(\mathbf{\Phi}, \mathbf{g}) = \min_{\mathbf{b} \succeq 0} \frac{1}{2} \| \mathbf{\Phi b} - \mathbf{g} \|_2^2 + \tau \mathcal{S}(\mathbf{b}), \tag{8}$$

then, we classify the video as

$$f_2(\mathbf{g}) = \big\{ j | \mathcal{R}\big(\mathbf{\Phi}^j, \mathbf{g}\big) < \mathcal{R}\big(\mathbf{\Phi}^i, \mathbf{g}\big), \quad j \neq i, (i, j) \in [1, \ldots, C] \big\}. \tag{9}$$

Here, we classify by selecting the class yielding a minimum reconstruction and complexity as given by $\mathcal{R}(\mathbf{\Phi}^j, \mathbf{g})$, corresponding to the energy associated to the jth class. Notice that in this procedure only a single vector $\mathbf{g}$ in $\Re_+^C$ needs to be sparsely represented for the whole video being classified, which is computationally very cheap of course.

3.4 Comparison of Representations for Classification

The bag-of-features approach is one of the most widely used techniques for action classification. It basically consists of applying K-means clustering to find K centroids, i.e., visual words, that are representative of all the training samples. Then, a video is represented as a histogram of visual word occurrences, by assigning one of the centroids to each of the extracted features in the video using (most often) Euclidean distance. These K centroids are found using a randomly selected subset of features coming from all the training data. While this has the advantage of not having to learn C subproblems, it is not explicitly exploiting/modeling label information available in the given supervised setting. Therefore, it is difficult to interpret directly the class relationships in these global, high dimensional histograms (K is usually in the 3,000–4,000 range). In addition, the visual words expressed as histograms equally weight the contribution from the data samples, regardless of how far these are from the centroids. For example, an extracted descriptor or feature from the data that does not correspond to any of the classes (e.g., background), will be assigned to one of the K centroids in the same manner as a descriptor that truly pertains to a class. Therefore, unless a robust metric is used, further increasing the computational complexity of the methods, this has the disadvantage of not properly accounting for outliers and could significantly disrupt the data distribution. In the proposed method, each of the data samples is represented as a sparse linear combination of dictionary atoms, hence represented from union of subspaces. Instead of representing an extracted feature with its closest centroid, it is represented by a *weighted* combination of atoms, thus better managing outliers. Analogue to a Mixture of Gaussians (MoG), the bag-of-features representation can be considered as a hard-thresholded MoG, where only one Gaussian distribution is allowed per sample, and its associated weight equals to one.

The learning process at the first level of the proposed model uses samples (vectorized spatio-temporal patches) from each action independently (in contrast to learning a global dictionary), and later encodes them as linear combinations of the learned dictionary atoms from all classes, where the class contribution is explicitly given in the obtained sparse codes. Since each data sample from a specific class can be represented by a different subset of dictionary atoms, the resulting sparse codes can have significant variations in the activation set. Sum-pooling in a per-class manner achieves invariance to the class subset (atom) selection. These sum-pooled vectors are used to quantify the association of the samples with each class (activity), and a significant dimensionality reduction is obtained by mapping these codes into a C-dimensional space (in contrast to performing explicit dimension reduction as in some of the techniques described above). We learn all the representations in a non-negative fashion. This is done for two reasons. First, we use the absolute value of the temporal gradient (to allow the same representation for samples with opposite contrast), so

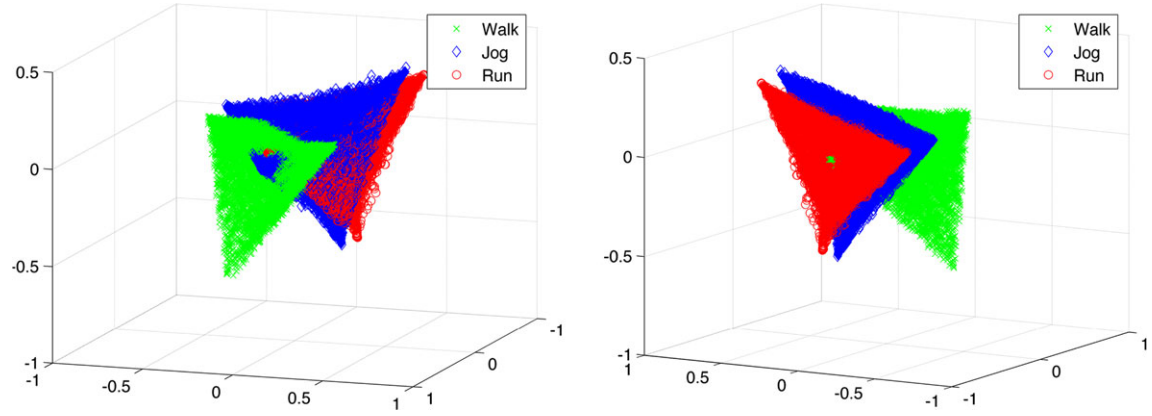

Fig. 2 Front and rear views of the first three principal components corresponding to the per-class ℓ_1-norm of data samples (using all the training videos from the KTH dataset) after the first level of sparse coding in our algorithm. The samples in *green* correspond to the *walk* class, the samples in *blue* correspond to the *jog* class, and the samples in *red* correspond to the *run* class (Color figure online)

all data values are nonnegative. Second, each data sample is normalized to have unit magnitude. After the per-class sum-pooling, this allows a mapping that is close to a probability space (the ℓ_1 norm of the sparse codes will be close to one). Therefore, the coefficients associated with each class give a good notion of the probability of each class in the extracted features.

Consider the example illustrated in Fig. 2. Shown are the first three principal components of all the C-dimensional sum-pooled vectors corresponding to the *jog*, *run*, and *walk* actions from the KTH dataset (details on this standard dataset will be presented in the experimental section). As we can see, some of the data points from each class intersect with the other two classes, corresponding to shared movements, or spatio-temporal structures that may well live in any of the classes' subspaces, a per-sample effect which we call *action mixtures*. Also, the actions have a global structure and position relative to each other within the $3D$ spatial coordinates, which appears to be related to the subjects' velocity (*jog* seems to be connected to *walk* and *run*). Therefore, this local characterization obtained at the first level, where the data points are mapped into a mixture space, indeed have a global structure. Thus, the purpose of the second level is to model an incoming video by taking into account its entire data distribution relative to this global structure, considering relationships between classes (actions), and expressing it sparsely using dictionary atoms that span the space of the individual actions. Such cross-action learning and exploitation is unique to the proposed model, when compared to those described above, and is achieved working on the natural low dimensional C-space, thereby being computationally very efficient.

4 Experimental Results

We evaluate the classification performance of the proposed method using 4 publicly available datasets: KTH, UT-Tower, UCF-Sports, and YouTube. The results presented include performance rates for each of the two levels of modeling, which we call SM-1 for the first level, and SM-2 for the second level. Separating both results will help in understanding the properties and capabilities of the algorithm in a per-level fashion. Remember that the additional computational cost of the second layer is basically zero, a simple sparse coding of a single low dimensional vector. Additionally, to illustrate the discriminative information available in the per-class sum-pooled vectors S, we include classification results of all datasets using a χ^2-kernel SVM in a one-against-the other approach, and we call this SM-SVM. In other words, the output from the first level of the proposed algorithm is the input to SM-SVM. For each classifier, we built the kernel matrix by randomly selecting 3,000 training samples. We report the mean accuracy after 1,000 runs. Finally, for comparison purposes, we include the best three performance rates reported in the literature. Often, these three are different for different datasets, indicating a lack of universality in the different algorithms reported in the literature (though often some algorithms are always close to the top, even if they do not make the top 3). Confusion matrices for SM-1 and SM-2 are also included for further analysis.

Table 1 shows the parameters used in SM-1 and SM-2 for each of the datasets in our experiments. The values were chosen so that good empirical results were obtained, but standard cross-validation methods can be easily applied to obtain optimal parameters. Note how we used the same basic parameters for all the very distinct datasets. The first three columns specify the amount of randomly selected spatio-temporal patches per video clip, the threshold used for interest point detection, and the size of the spatio-temporal over-

Table 1 Parameters for each of the datasets. The first three columns are related to feature extraction parameters. The last four columns specify sparse coding/dictionary-learning parameters

Dataset	Feature extraction			Sparse modeling			
	n/clip	η	m	λ	τ	k_j	l_j
KTH	30000/#clips	0.20	$15 \times 15 \times 9$	$20/\sqrt{m}$	$1/C$	768	32
UT-Tower	30000/#clips	0.10	$15 \times 15 \times 9$	$20/\sqrt{m}$	$1/C$	768	32
UCF-Sports	30000/#clips	0.20	$15 \times 15 \times 9$	$20/\sqrt{m}$	$1/C$	768	32
YouTube	40000/#clips	0.20	$15 \times 15 \times 9$	$20/\sqrt{m}$	$0.5/C$	768	64

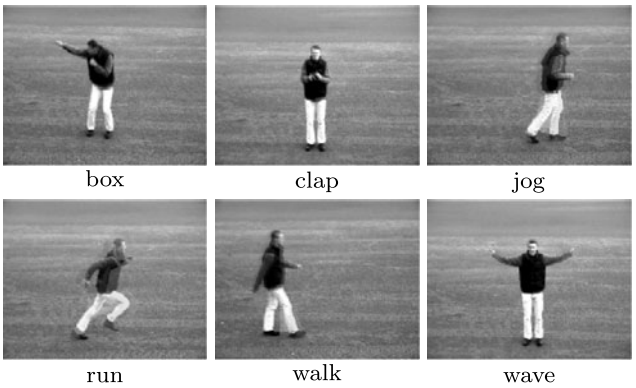

Fig. 3 Sample frames from the KTH dataset

lapping patches, respectively. The last four columns specify the sparsity parameters and the number of dictionary atoms used for SM-1 and SM-2 modeling, respectively. Note how for simplicity we also used same dictionary size for all classes. We now present the obtained results.

4.1 KTH

The KTH dataset[4] (Schuldt et al. 2004) is one of the most popular benchmark action data. It consists of approximately 600 videos of 25 subjects, each performing $C = 6$ actions: *box*, *clap*, *jog*, *run*, *walk*, and *wave*. Each of these actions were recorded at 4 environment settings: outdoors, outdoors with camera motion (zoom in and out), outdoors with clothing change, and indoors. We followed the experimental settings from Schuldt et al. (2004). That is, we selected subjects 11–18 for training and subjects 2–10, and 22 for testing (no training performed on this set). Figures 3 and 4 show sample frames from each of the actions and the learned dictionaries for both layers, respectively. Notice, Fig. 4, how the second level encodes the ℓ_1 energy distributions of each class with respect to the other classes.

Table 2 presents the corresponding results. We obtain 97.9 %, 94.4 % and 96.3 % with SM-SVM, SM-1 and SM-2, respectively. Confusion matrices for SM-1 and SM-2 are shown in Fig. 5. As expected, there is some misclassification error occurring between the *jog*, *run*, and *walk* actions, all which share most of the spatio-temporal structures. SM-2 performs better, since it combines all the local information with the global information from **S** and **g**, respectively. The three best performing previous methods are Wang et al. (2011) (94.2 %), Kovashka and Grauman (2010) (94.5 %), and Guo et al. (2010) (97.4 %). The method described in Wang et al. (2011) performs tracking of features using dense sampling. The method in Kovashka and Grauman (2010) requires bag-of-features using several detectors at several levels, dimensionality reduction with PCA, and also uses neighborhood information, which is much more sophisticated than our method. The closest result to our method is 97.4 %, described in Guo et al. (2010). Their method is similar in nature to ours, as it uses features derived from optical flow representing entire videos, further highlighting the need for global information for higher recognition. As mentioned before, there is no cross-class learning in such approach.

4.2 UT-Tower

The UT-Tower dataset[5] (Chen et al. 2010) simulates an "aerial view" setting, with the goal of recognizing human actions from low-resolution remote sensing (people's height is approximately 20 pixels on average), and is probably from all the tested datasets the most related to standard surveillance applications. There is also camera jitter and background clutter. It consists of 108 videos of 12 subjects, each performing $C = 9$ actions using 2 environment settings. The first environment setting is an outdoors concrete square, with the following recorded actions: *point, stand, dig*, and *walk*. In the second environment setting, also outdoors, the following actions were recorded: *carry*, *run*, *wave with one arm* (*wave1*), *wave with both arms* (*wave2*), and *jump*. We converted all the frames to grayscale values. A set of automatically detected bounding box masks centered at each subject are provided with the data, as well as a set of automatically detected tracks for each subject. We used the set of bounding box masks but not the tracks. All results follow the standard for this dataset Leave One Out Cross Validation (LOOCV) procedure. Figure 6 shows sample frames for each action.

Table 3 presents the results. We obtained 98.1 %, 97.2 %, and 100 % for SM-SVM, SM-1, and SM-2, respectively.

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

[5] http://cvrc.ece.utexas.edu/SDHA2010/Aerial_View_Activity.html.

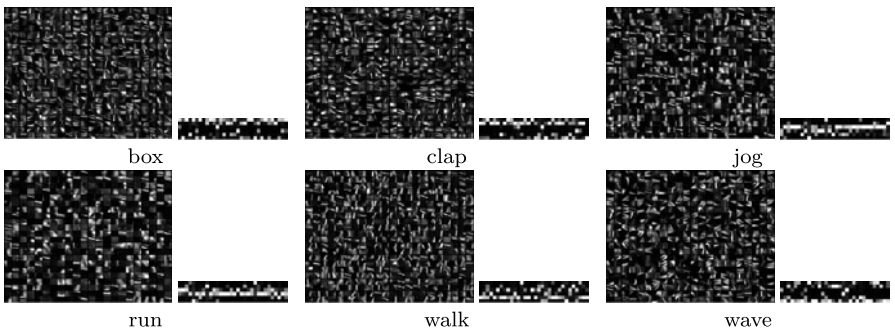

Fig. 4 Learned action dictionaries from the KTH dataset for both levels

Table 2 Results for the KTH dataset

Method	Overall accuracy (%)
Wang et al. (Wang et al. 2011)	94.2
Kovashka et al. (Kovashka and Grauman 2010)	94.5
Guo et al. (Guo et al. 2010)	97.4
SM-SVM	**97.9**
SM-1	94.4
SM-2	96.3

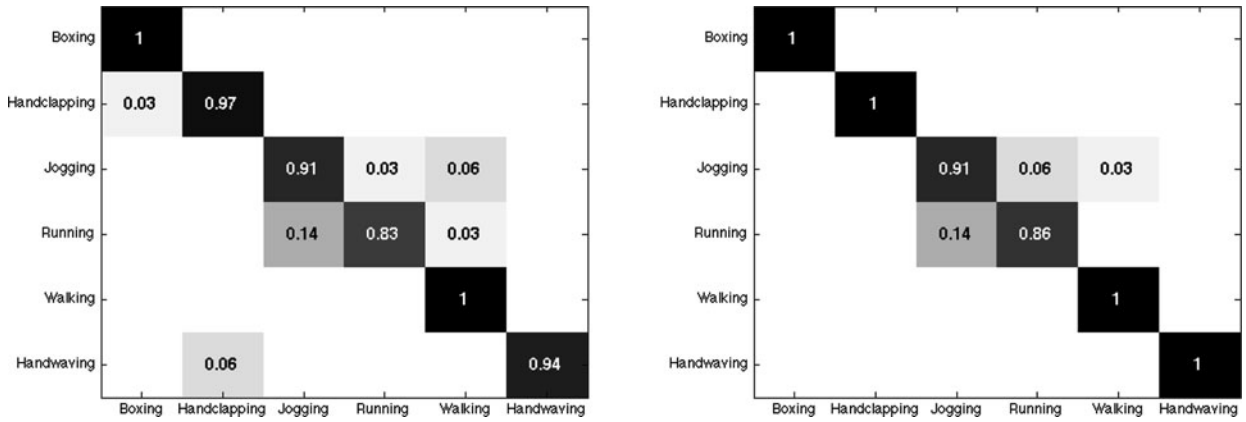

Fig. 5 Confusion matrices from classification results on the KTH dataset using SM-1 and SM-2. The value on each cell represents the ratio between the number of samples labeled as the column's label the total number of samples corresponding to the row's label

Fig. 6 Sample frames from the UT-Tower dataset

The only confusion in SM-1 occurs between the *point* and *stand* classes and between the *wave1* and *wave2* classes (see Fig. 7), since there are evident action similarities between these pairs, and the low resolution in the videos provides a low amount of samples for training. The methods proposed in Vezzani et al. (2010) and Gall et al. (2011) both obtained 93.9 %. In Vezzani et al. (2010), the authors use a Hidden Markov Model (HMM) based technique with bag-of-features from projected histograms of extracted foreground. The method in Gall et al. (2011) uses two stages of random forests from features learned based on Hough transforms. The third best result was obtained with the method in Guo et al. (2010) as reported in Ryoo et al. (2010). Again, our method outperforms the other methods with a simpler approach.

4.3 UCF-Sports

The UCF-Sports dataset[6] (Rodriguez et al. 2008) consists of 150 videos acquired from sports broadcast networks. It has $C = 10$ action classes: *dive, golf swing, kick, weight-lift, horse ride, run, skateboard, swing (on a pommel horse*

[6] http://server.cs.ucf.edu/~vision/data.html#UCFSportsActionDataset.

Table 3 Results for the UT-Tower dataset

Method	Overall accuracy (%)
Guo et al. (Guo et al. 2010; Ryoo et al. 2010)	97.2
Vezzani et al. (Vezzani et al. 2010)	93.9
Gall et al. (Gall et al. 2011)	93.9
SM-SVM	98.1
SM-1	97.2
SM-2	**100**

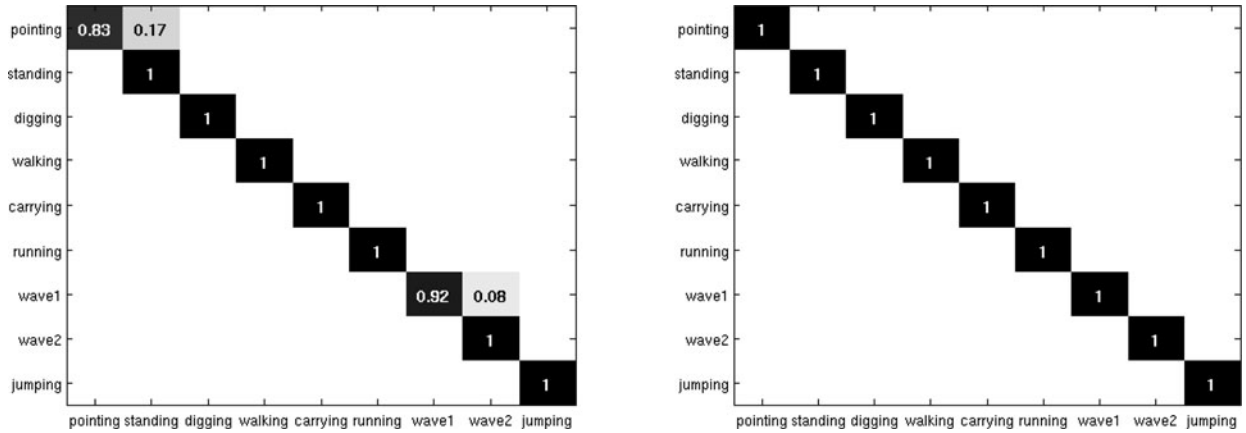

Fig. 7 Confusion matrices from classification results on the UT-Tower dataset using SM-1 and SM-2

Fig. 8 Sample frames from the UCF-Sports dataset

and on the floor), swing (on a high bar), and *walk*. Figure 8 shows sample frames for each action. This dataset has camera motion and jitter, highly cluttered and dynamic backgrounds, compression artifacts, and variable illumination settings at variable spatial resolution, and 10 fps. We followed the experimental procedure from Wang et al. (2009), which uses LOOCV. Also as in Wang et al. (2009), we extended the dataset by adding a flipped version of each video with respect to its vertical axis, with the purpose of increasing the amount of training data (while the results of our algorithm are basically the same without such flipping, we here preformed it to be compatible with the experimental settings in the literature). These flipped versions were only used during the training phase. All videos are converted to gray level for processing. We also used the spatial tracks provided with the dataset for the actions of interest.

Classification results are presented in Table 4, and we show the SM-1 and SM-2 confusion matrices in Fig. 9. We obtained 94.7 %, 96.0 %, and 97.3 % overall classification rates with SM-SVM, SM-1, and SM-2, respectively. In this case, all three SM methods achieve higher classification accuracies than those previously reported in Kovashka and Grauman (2010), Le et al. (2011), and Wang et al. (2011). We observe misclassification errors in the *run* and *horse ride* classes for the SM-1, and are alleviated by SM-2.

4.4 YouTube

The YouTube Dataset[7] (Liu et al. 2009) consists of 1,168 sports and home videos from YouTube with $C = 11$ types of actions: *basketball shooting*, *cycle*, *dive*, *golf swing*, *horse back ride*, *soccer juggle*, *swing*, *tennis swing*, *trampoline jump*, *volleyball spike*, and *walk with a dog*. Each of the action sets is subdivided into 25 groups sharing similar environment conditions. Similar to the UCF-Sports dataset, this is a more challenging dataset with camera motion and jitter,

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

Table 4 Results for the UCF-Sports dataset

Method	Overall accuracy (%)
Le et al. (Le et al. 2011)	86.5
Wang et al. (Wang et al. 2011)	88.2
Kovashka et al. (Kovashka and Grauman 2010)	87.5
SM-SVM	94.7
SM-1	96.0
SM-2	**97.3**

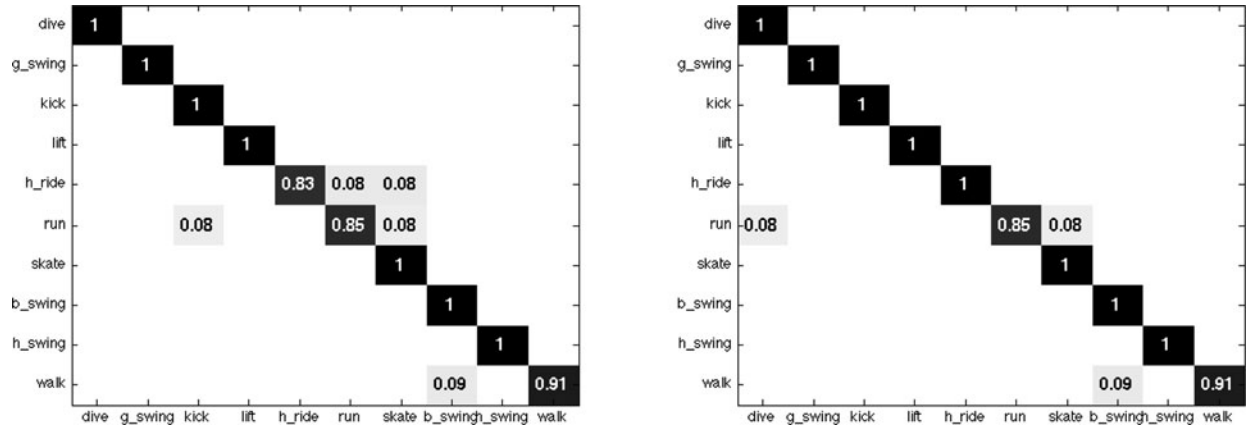

Fig. 9 Confusion matrices from classification results on the UCF-Sports dataset using SM-1 and SM-2

Fig. 10 Sample frames from the YouTube dataset

highly cluttered and dynamic backgrounds, compression artifacts, and variable illumination settings. The spatial resolution is 320 × 240 at variable 15–30 fps. We followed the experimental procedure from Liu et al. (2009), that is, a group-based LOOCV, where training per action is based on 24 out of 25 of the groups, and the remaining group is used for classification. We also converted all frames to grayscale values. Figure 10 shows sample frames from each action.

Table 5 shows the overall classification results of our proposed method and comparisons with the state of the art methods, and Fig. 11 shows the confusion matrices corresponding to SM-1 and SM-2. We obtain overall classification rates of 83.8 %, 86.3 %, and 89.5 % from SM-SVM, SM-1, and SM-2, respectively.

The accuracy attained by SM-SVM is in the same ballpark as the best reported results using dense trajectories, which again incorporates dense sampling at multiple spatio-temporal scales using more sophisticated features, in addition to tracking. Again, the global *and* local nature of SM-2 greatly helps to achieve the highest accuracy, as it decreased the scattered instances of misclassification obtained by SM-1 by implicitly imposing sparsity in a grouping fashion.

4.5 Computation Time

The computational efficiency of the model comes from performing simple temporal gradient and thresholding operations in the feature extraction step, and simple decision rules from classification that come directly from the sparse coding of the data. The experiments were conducted on an Intel Core 2 Duo (2.53 GHz) with 4 GB of memory using MAT-

Table 5 Results for the YouTube dataset

Method	Overall accuracy (%)
Le et al. (Le et al. 2011)	75.8
Wang et al. (Wang et al. 2011)	84.2
Ikizler-Cinbis et al. (Ikizler-Cinbis and Sclaroff 2010)	75.2
SM-SVM	83.8
SM-1	86.3
SM-2	**89.5**

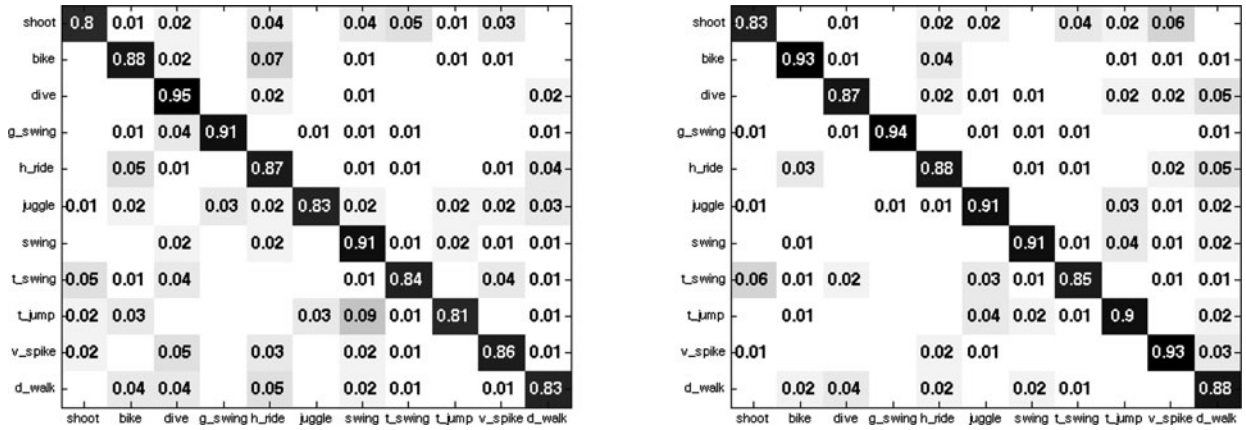

Fig. 11 Confusion matrices from classification results on the YouTube dataset using SM-1 and SM-2

LAB. Using the training videos in Schuldt et al. (2004) from the KTH dataset, it took 4,064 seconds overall, that is, feature extraction, learning, and classification. The most time consuming part was the sparse coding, with 2045 seconds, followed by dictionary learning, with 993 seconds in total. The overall feature extraction procedure took 441 seconds. Testing on a single $120 \times 160 \times 100$ video, it took a total of 17 seconds, where 3.39 seconds correspond to feature extraction, and 11.90 seconds correspond to classification, thus taking approximately 15 seconds of computation overall, or 6.7 frames per second.

4.6 Summary

Summarizing these results, we reported an increase in the classification accuracy of 0.5 % in KTH, 2.8 % in UT-Tower, 9.1 % in UCF-Sports, and 5.3 % in YouTube. While the prior state-of-the-art results where basically obtained with a variety of algorithms, our proposed framework uniformly outperforms all of them without per-dataset parameter tuning, and often with a significantly simpler modeling and classification technique. These results clearly show that the dimension reduction attained from $\mathbf{A}$ to $\mathbf{S}$ and the local to global mapping do not degrade the discriminative information, but on the contrary, they enhance it.

To further stress the generality of our algorithm, we have not tuned parameters for any of the datasets. Some parameters though could be adapted to the particular data, e.g., the patch size should be adapted to the spatial and temporal resolution of the videos if taken from the same camera.

Following the simplicity of the framework here presented, one might be tempted to go even simpler. For example, we could consider replacing the learned dictionaries by simpler vector-quantization. We have investigated that and obtained that for example, for the UCF-Sports dataset, the results are significantly worse, attaining a classification accuracy of 18 %.

Finally, we have observed that the natural nonnegativity constraint often improves the results, although sometimes the improvement is minor, and as a consequence, we opted to leave it as part of the framework.

5 Concluding Remarks

We presented a two-level hierarchical sparse model for the modeling and classification of human actions. We showed how modeling local and global observations using concepts of sparsity and dictionary learning significantly improves classification capabilities. We also showed the generality of the algorithm to tackle problems from multiple diverse publicly available datasets: KTH, UT-Tower, UCF-Sports, and YouTube, with a relatively small set of parameters (uniformly set for all the datasets), a single and simple feature, and a single spatio-temporal scale.

Although simple in nature, the model gives us insight into new ways of extracting highly discriminative information directly from the combination of local and global sparse coding, without the need of explicitly incorporating discriminative terms in the optimization problem and without the need to manually design advanced features. In fact, the results from our experiments demonstrate that the sparse coefficients that emerge from a multi-class structured dictionary are sufficient for such discrimination, and that even with a simple feature extraction/description procedure, the model is able to capture fundamental inter-class distributions.

The model's scalability could become a challenge when the number of classes is very large, since it will significantly increase the size of the dictionary. In such case, it would be useful to integrate newly emerging algorithms for fast sparse approximations such as those proposed by Gregor and LeCun (2010) and Xiang et al. (2011), hence rendering the model more efficient. We are also interested in incorporating locality to the model, which could provide additional insight for analyzing more sophisticated human interactions. In addition, using a combination of features (e.g., multiscale) as the learning premise would help in dealing with much more complex data acquisition effects such as multi-camera shots and rapid scale variations such as those present in the Hollywood-2 human actions dataset (Marszałek et al. 2009). We are also exploiting time-dependencies for activity-based summarization of motion imagery.

Acknowledgements Work partially supported by NGA, ONR, ARO, NSF, Level Sets Systems, and AFOSR (NSSEFF). The authors would like to thank Pablo Sprechmann, Dr. Mariano Tepper, and David S. Hermina for very helpful suggestions and insightful discussions. We also thank Dr. Julien Mairal for providing publicly available sparse modeling code (SPAMS http://www.di.ens.fr/willow/SPAMS/downloads.html) used in this work.

References

Aharon, M., Elad, M., & Bruckstein, A. (2006). K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, *54*(11), 4311–4322.

Blake, R., & Shiffrar, M. (2007). Perception of human motion. *Annual Review of Psychology*, *58*(1), 47–73.

Bruckstein, A., Donoho, D., & Elad, M. (2009). From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, *51*(1), 34–81.

Cadieu, C., & Olshausen, B. A. (2008). Learning transformational invariants from natural movies. In *NIPS* (pp. 209–216).

Castrodad, A., Xing, Z., Greer, J., Bosch, E., Carin, L., & Sapiro, G. (2011). Learning discriminative sparse representations for modeling, source separation, and mapping of hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, *49*(11), 4263–4281.

Charles, A., Olshausen, B., & Rozell, C. (2011). Learning sparse codes for hyperspectral imagery. *IEEE Journal of Selected Topics in Signal Processing*.

Chen, C., Ryoo, M. S., & Aggarwal, J. K. (2010). UT-Tower dataset: aerial view activity classification challenge. http://cvrc.ece.utexas.edu/SDHA2010/Aerial_View_Activity.html.

Dalal, N., & Triggs, B. (2006). Human detection using oriented histograms of flow and appearance. In *ECCV*.

Dean, T., Washington, R., & Corrado, G. (2009). Recursive sparse, spatiotemporal coding. In *ISM* (pp. 645–650).

Dollar, P., Rabaud, V., Cottrell, G., & Belongie, S. (2005). 2nd joint IEEE international workshop on behavior recognition via sparse spatio-temporal features. In *Visual surveillance and performance evaluation of tracking and surveillance* (pp. 65–72).

Donoho, D. L. (2000). High-dimensional data analysis: the curses and blessings of dimensionality. In *American Mathematical Society conference math challenges of the 21st century*.

Gall, J., Yao, A., Razavi, N., van Gool, L., & Lempitsky, V. (2011). Hough forests for object detection, tracking, and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(11), 2188–2202.

Gregor, K., & LeCun, Y. (2010). Learning fast approximations of sparse coding. In *ICML* (pp. 399–406).

Guo, K., Ishwar, P., & Konrad, J. (2010). Action recognition using sparse representation on covariance manifolds of optical flow. In *AVSS* (pp. 188–195).

Ikizler-Cinbis, N., & Sclaroff, S. (2010). Object, scene and actions: combining multiple features for human action recognition. In *ECCV* (pp. 494–507).

Jhuang, H., Serre, T., Wolf, L., & Poggio, T. (2007). A biologically inspired system for action recognition. In *ICCV* (pp. 1–8).

Kläser, A., Marszałek, M., & Schmid, C. (2008). A spatio-temporal descriptor based on 3d-gradients. In *BMVC*.

Kovashka, A., & Grauman, K. (2010). Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *CVPR* (pp. 2046–2053).

Laptev, I., & Lindeberg, T. (2003). Space-time interest points. In *ICCV* (pp. 432–439).

Laptev, I., Marszałek, M., Schmid, C., & Rozenfeld, B. (2008). Learning realistic human actions from movies. In *CVPR*.

Le, Q., Zou, W., Yeung, S., & Ng, A. (2011). Learning hierarchical spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*.

Liu, J., Luo, J., & Shah, M. (2009). Recognizing realistic actions from videos "in the wild". In *CVPR*.

Mairal, J., Bach, F., Ponce, J., Sapiro, G., & Zisserman, A. (2008). Supervised dictionary learning. In *NIPS* (pp. 1033–1040).

Mairal, J., Bach, F., Ponce, J., & Sapiro, G. (2010). Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, *11*, 19–60.

Marszałek, M., Laptev, I., & Schmid, C. (2009). Actions in context. In *CVPR*.

Ramirez, I., Sprechmann, P., & Sapiro, G. (2010). Classification and clustering via dictionary learning with structured incoherence and shared features. In *CVPR* (pp. 3501–3508).

Rodriguez, M. D., Ahmed, J., & Shah, M. (2008). Action Mach: a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*.

Ryoo, M., Chen, C., Aggarwal, J., & Chowdhury, R. A. (2010). An overview of contest on semantic description of human activities (sdha) 2010. In *ICPR-contests* (pp. 270–285).

Schuldt, C., Laptev, I., & Caputo, B. (2004). Recognizing human actions: a local svm approach. In *ICPR* (pp. 32–36).

Scovanner, P., Ali, S., & Shah, M. (2007). A 3-dimensional sift descriptor and its application to action recognition. In *ACM multimedia* (pp. 357–360).

Shao, L., & Mattivi, R. (2010). Feature detector and descriptor evaluation in human action recognition. In *CIVR* (pp. 477–484).

Sprechmann, P., & Sapiro, G. (2010). Dictionary learning and sparse coding for unsupervised clustering. In *ICASSP*.

Taylor, G., Fergus, R., Le Cun, Y., & Bregler, C. (2010). Convolutional learning of spatio-temporal features. In *ECCV* (pp. 140–153).

Tibshirani, R. (1994). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, *58*, 267–288.

Vezzani, R., Davide, B., & Cucchiara, R. (2010). HMM based action recognition with projection histogram features. In *ICPR* (pp. 286–293).

Wang, H., Ullah, M. M., Kläser, A., Laptev, I., & Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. In *BMVC*.

Wang, H., Kläser, A., Schmid, C., & Cheng-Lin, L. (2011). Action recognition by dense trajectories. In *CVPR* (pp. 3169–3176).

Willems, G., Tuytelaars, T., & van Gool, L. (2008). An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV* (pp. 650–663).

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., & Ma, Y. (2008). Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *31*(2), 210–227.

Xiang, Z., Xu, H., & Ramadge, P. (2011). Learning sparse representations of high dimensional data on large scale dictionaries. In *NIPS* (pp. 900–908).

Yeffet, L., & Wolf, L. (2009). Local trinary patterns for human action recognition. In *ICCV* (pp. 492–497).