SPARSE MODELING OF HUMAN ACTIONS FROM MOTION IMAGERY

By

Alexey Castrodad

and

Guillermo Sapiro

IMA Preprint Series # 2378

(September 2011)

INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA
400 Lind Hall
207 Church Street S.E.
Minneapolis, Minnesota 55455–0436

Phone: 612-624-6066 Fax: 612-626-7370
URL: http://www.ima.umn.edu

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **02 SEP 2011**	2. REPORT TYPE	3. DATES COVERED **00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE **Sparse Modeling of Human Actions from Motion Imagery**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Minnesota,Institute for Mathematics and Its Applications,207 Church Street SE,Minneapolis,MN,55455-0436**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

An efficient sparse modeling pipeline for the classification of human actions from video is here developed. Spatio-temporal features that char- acterize 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 com- bination 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 outper- forms previously reported results for virtually all standard datasets.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**26**	

Sparse Modeling of Human Actions from Motion Imagery

Alexey Castrodad and Guillermo Sapiro [*]

September 2, 2011

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.

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.

[*] The authors are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, 55455 USA e-mail: {castr103, guille}@umn.edu. Alexey Castrodad is also with the Department of Defense.

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 [11], and will be key to overcoming 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 [2]. 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, and accounts for interclass relationships using global and local perspectives. As illustrated in Section 2 (this section also briefly describes prior art), and described in detail in Section 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, only *one feature* and *one sampling scale* are sufficient for highly accurate activity classification. We claim that there is a great deal of information inherent in the sparse representations that have not yet been fully explored. In [23] 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 and more moderate computational cost. In Section 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 Section 5.

2 Problem Statement and Overview of the Proposed Framework

In this section we define the problem and present an overview of the proposed framework. We also discuss the recent related literature and compare our proposed framework to the standard *bag-of-features* approach. Details on the introduced model will be presented in the next section.

Assume we have a set of labeled videos, each containing 1 of C known actions (classes) with associated label $j \in [1, 2, ..., C]$.[1] 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 Figure 1.

[1] 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.

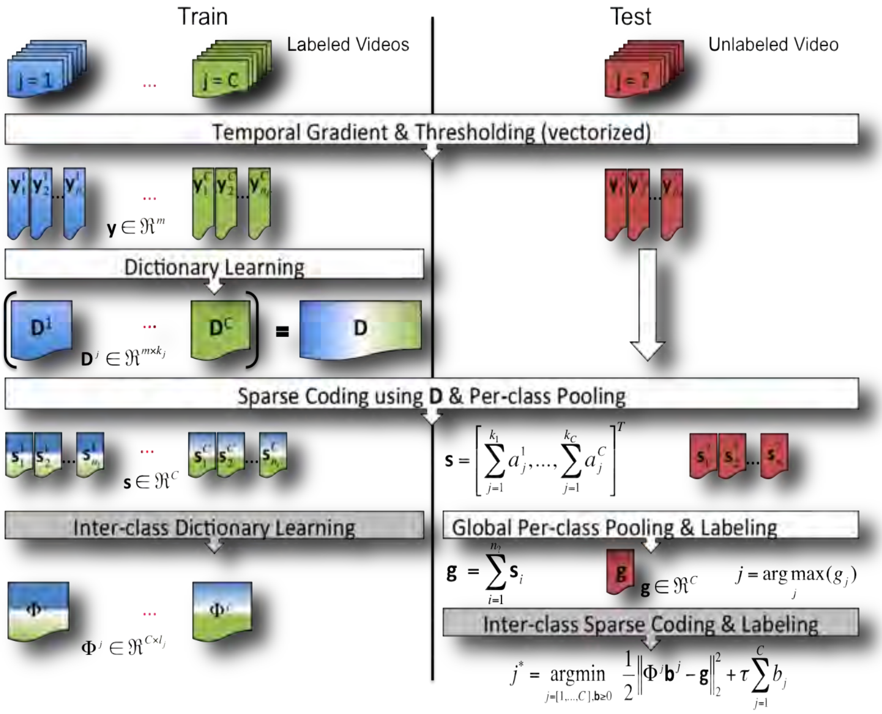

Train
Test
Labeled Videos
Unlabeled Video
j = 1
j = C
j = ?
Temporal Gradient & Thresholding (vectorized)
$\mathbf{y} \in \Re^m$
Dictionary Learning
$\mathbf{D}^1$
$\mathbf{D}^C$
$=$
$\mathbf{D}$
$\mathbf{D}^j \in \Re^{m \times k_j}$
Sparse Coding using D & Per-class Pooling
$\mathbf{s} = \left[\sum_{j=1}^{k_1} a_j^1, \ldots, \sum_{j=1}^{k_C} a_j^C\right]^T$
$\mathbf{s} \in \Re^C$
Inter-class Dictionary Learning
Global Per-class Pooling & Labeling
$\mathbf{g} = \sum_{i=1}^{n_j} \mathbf{s}_i$
$\mathbf{g} \in \Re^C$
$j = \arg\max_j (g_j)$
$\Phi^j \in \Re^{C \times l_j}$
Inter-class Sparse Coding & Labeling
$j^* = \underset{j=[1,\ldots,C],\mathbf{b} \geq 0}{\operatorname{argmin}} \frac{1}{2}\|\Phi^j \mathbf{b}^j - \mathbf{g}\|_2^2 + \tau \sum_{j=1}^{C} b_j$

Figure 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. (This is a color figure.)

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. 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 each of the dictionaries $\mathbf{\Phi}^j$. The video's label j^* is selected such that the representation obtained with the $j-th$ action dictionary $\mathbf{\Phi}^j$ yields the minimum sparsity *and* reconstruction trade-off.

Section 3 will provide details on the just described algorithm, but before that, with this overview of our proposed modeling framework in mind, let us comment on recent related work.

2.1 Related Work and Analysis

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 [10], Harris3D [18], Hessian [36], and dense sampling[2] [12, 20, 34]. 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 [28], the Cuboids feature [10], Histograms of Oriented Gradients (HOGs) [19], and its extension to the temporal domain, i.e., HOG3D [16], combinations of HOG and Histograms of Optical Flow (HOF) [19], Extended Speeded Up Robust Features (ESURF), Local Trinary Patterns [38], and Motion Boundary Histograms (MBH) [8]. 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 refer to [29, 35] 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 [35], 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

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

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 [37] (without dictionary learning), digit and texture classification [23, 24], hyperspectral imaging [5, 6], among numerous other applications. It has also been applied recently for motion imagery analysis for example in [4, 9, 13, 31]. In [9], 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 [13], 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 [13] 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 [13] 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 [15], 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 [31], 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 C 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 [20], 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 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 [15, 20, 31]. 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,[3] 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 [9, 31]. 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).

2.1.1 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

[3] 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.

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 sub-problems, 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. In contrast, our representation is able to better adapt to the data structure, with no prior hard assumption on its distribution (does not assumes a MoG).

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). As we will see in Section 3.2, we learn all the representations in a nonnegative 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 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

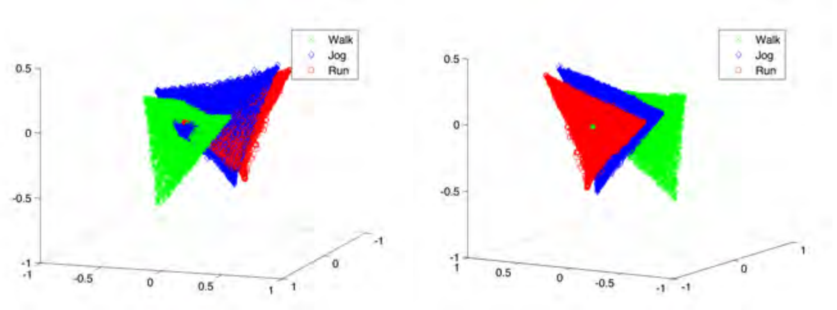


Figure 2: Front and rear views of the first three principal components corresponding to the per-class ℓ_1-norm of data samples (using all the training videos from the KTH dataset) after the first level of sparse coding in our algorithm. The samples in green correspond to the *walk* class, the samples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

features.

Consider the example illustrated in Figure 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.

To recap, the proposed model, described in detail next, is significantly simpler than previously proposed ones, both at the concept level and at the computational cost one, still achieving state-of-the-art results for virtually all standard popular datasets.

3 Sparse Modeling for Action Classification

We now give a detailed description of the proposed modeling and classification algorithm for activity recognition. 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 the section with a description of the labeling/classification procedure.

3.1 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 $\mathbf{y}(p) = vect(|\mathbf{I}_t(p)|)$ 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 $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_n] \in \Re^{m \times n}$, 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:

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

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 D. Under assumptions on the sparsity of the signal and the structure of the dictionary $\mathbf{D}$ (see [3]), there exists $\lambda > 0$ such that (1) is equivalent to solving

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

known as the Lasso [32]. 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 $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_n] \in \Re^{k \times n}$, 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

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

which can be efficiently solved using algorithms like the K-SVD [1, 22].

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 Learning Action-specific Dictionaries

Since we are in the supervised setting, there are labeled training data available for each of the actions. Let $\mathbf{Y}^j = [\mathbf{y}_1^j, ..., \mathbf{y}_{n_j}^j] \in \Re^{m \times n_j}$ be the n_j extracted samples corresponding to the $j-th$ action/class. We obtain the $j-th$ action representation (class-specific dictionary) $\mathbf{D}^j \in \Re_+^{m \times k_j}$ by solving

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

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.

3.3 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 $\mathbf{D} = [\mathbf{D}^1, ..., \mathbf{D}^C] \in \Re_+^{m \times k}$, where $k = \sum_{j=1}^{C} k_j$. Then we get, for the entire data being processed Y,

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

where $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_n] \in \Re_+^{k \times n}$, $\mathbf{a}_i = [a_i^1, ..., a_i^{k_1}, ..., a_i^{k_C}]^T \in \Re_+^k$, 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 sub-dictionary (see also [37] for a related coding strategy for faces). That is, each data sample becomes a "mixture" of the actions modeled in $\mathbf{D}$, and the component (or fraction) of the $j-th$ 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 $j-th$ action are shared, then the contribution from the other action representations will be zero, meaning that the data sample is purely pertaining of the $j-th$ 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 $\mathbf{s} = [\mathcal{S}(\mathbf{a}^1), ..., \mathcal{S}(\mathbf{a}^C)]^T \in \Re_+^C$ be the per-class ℓ_1-norm vector corresponding to the data sample $\mathbf{y}$, and letting $\mathbf{S} = [\mathbf{s}_1, ..., \mathbf{s}_n] \in \Re_+^{C \times n}$ 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.

3.4 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:

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

where $\mathbf{B}^j = [\mathbf{b}_1^j, ..., \mathbf{b}_{n_j}^j] \in \Re^{l_j \times n_j}$ 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.

3.5 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.

3.5.1 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

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

This classification, already provides competitive results, especially with actions that do not share too many spatio-temporal structures, see Section4. 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.

3.5.2 Labeling After Level 2

There are cases where there are known shared (local) movements between actions (e.g., running and jogging), 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 [30], we develop a classification scheme for the second level. Let

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

then, we classify the video as

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

Here, we classify by selecting the class yielding a minimum reconstruction and complexity as given by $\mathcal{R}(\mathbf{\Phi}^j, \mathbf{g})$. 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.

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. 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 overlapping 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.

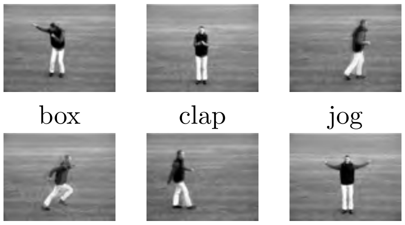
4.1 KTH

The KTH dataset[4] [27] 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 kept the original spatial

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

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	0.15	$15 \times 15 \times 7$	$15/\sqrt{m}$	$1/C$	512	32
UT-Tower	30000/#clips	0.15	$15 \times 15 \times 7$	$15/\sqrt{m}$	$1/C$	512	32
UCF-Sports	20000/#clips	0.15	$15 \times 15 \times 7$	$15/\sqrt{m}$	$1/C$	512	32
YouTube	40000/#clips	0.15	$15 \times 15 \times 7$	$15/\sqrt{m}$	$1.5/C$	512	128



box clap jog

run walk wave

Figure 3: Sample frames from the KTH dataset.

and temporal resolution, 120 × 160 at 25 frames per second (fps) and followed the experimental settings from [27]. That is, we selected subjects $11 - 18$ for training and subjects $2 - 10$, and 22 for testing (the validation subset was not used). Figure 3 shows sample frames from each of the actions.

Table 2 presents the corresponding results. We obtain 87.6%, 88.8% and 100% with SM-SVM, SM-1 and SM-2, respectively. Confusion matrices for SM-1 and SM-2 are shown in Figure 4. As expected, for SM-1 there is some misclassification error occurring between the *jog, run,* and *walk* actions, all which share most of the spatio-temporal structures. This illustrates why SM-2 performs significantly better, since it combines all the local information with the global information from $\mathbf{S}$ and $\mathbf{g}$, respectively. The three best performing previous methods are [34] (94.2%), [17] (94.5%), and [13] (97.4%). The method described in [34] performs tracking of features using dense sampling. The method in [17] 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 [13]. 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.

Table 2: Results for the KTH dataset.

Method	**Overall Accuracy (%)**
Wang *et al.* [34]	94.2
Kovashka *et al.* [17]	94.5
Guo *et al.* [13]	97.4
SM-SVM	87.6
SM-1	88.8
SM-2	**100**

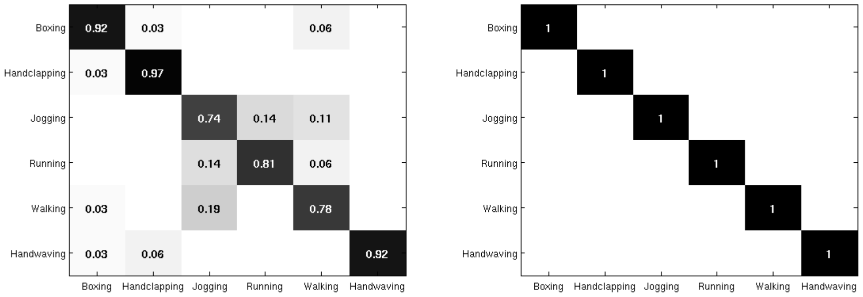


Figure 4: Confusion matrices from classification results on the KTH dataset using SM-1 and SM-2. The value on each cell represents the ratio between the number of samples labeled as the column's label the total number of samples corresponding to the row's label.

4.2 UT-Tower

The UT-Tower dataset[5] [7] 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 kept the original resolution of 320 × 240 at 10 fps, and 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

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

Figure 5: Sample frames from the UT-Tower dataset.

(LOOCV) procedure. Figure 5 shows sample frames for each action.

Table 3 presents the results. We obtained 93.3%, 97.2%, and 100% for SM-SVM, SM-1, and SM-2, respectively. The only confusion in SM-1 occurs between the *point* and *stand* classes and between the *wave1* and *wave2* classes (see Figure 6), 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 [33] and [12] both obtained 93.9%, which is comparable to the SM-SVM results. In [33], the authors use a Hidden Markov Model (HMM) based technique with bag-of-features from projected histograms of extracted foreground. The method in [12] uses two stages of random forests from features learned based on Hough transforms. The third best result was obtained with the method in [13] as reported in [26]. Again, our method outperforms the other methods with a simpler approach.

Table 3: Results for the UT-Tower dataset.

Method	**Overall Accuracy (%)**
Guo *et al.* [13, 26]	97.2
Vezzani *et al.* [33]	93.9
Gall *et al.* [12]	93.9
SM-SVM	93.3
SM-1	97.2
SM-2	**100**

4.3 UCF-Sports

The UCF-Sports dataset[6] [25] 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 and on the floor), swing (on a high bar)*, and *walk*. 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 [35], which uses LOOCV, and re-sampled the videos to half the spatial resolution. Also as in [35], we extended the dataset by adding a flipped version of each video with respect to its vertical axis, with the purpose

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

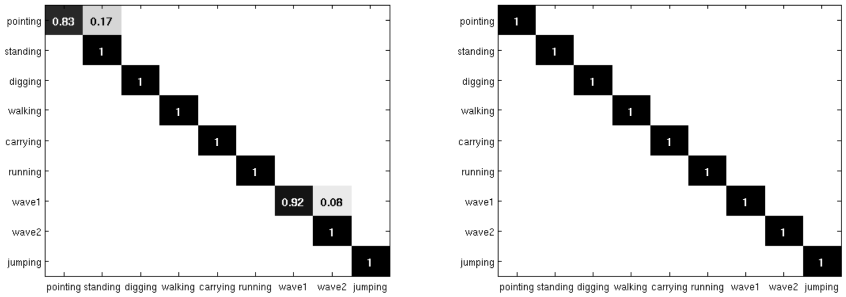


Figure 6: Confusion matrices from classification results on the UT-Tower dataset using SM-1 and SM-2.

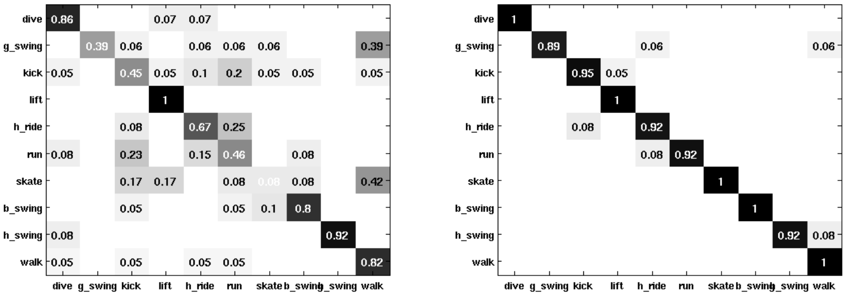
dive golf swing kick weight lift horse ride

run skateboard swing high bar swing walk

Figure 7: Sample frames from the UCF-Sports dataset.

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. While the dataset includes spatial tracks for the actions of interest, in keeping with the concept of as simple as possible pre-processing, these were not used in our experiments. Figure 7 shows sample frames from each action.

Classification results are presented in Table 4, and we show the SM-1 and SM-2 confusion matrices in Figure 8. We obtained 87.6%, 66.3%, and 96.0% overall classification rates with SM-SVM, SM-1, and SM-2, respectively. Clearly, the methods from [17, 20, 34] outperformed SM-1, and are in the same ballpark as SM-SVM, while still the best performance is obtained for the proposed SM-2. SM-1 failed to properly recognize the *skate* action (8%), and also performed poorly in recognizing *golf swing* (39%) and *kick* (45%), which caused confusion in recognizing the *run* action (46%). There are a number of possible reasons for this. The *skate* action has significant misclassification errors from the *walk* and *kick* classes. The front side angle of the camera when shooting makes it difficult to capture changes in velocity, and the movements of the skater while putting a foot in the ground for propulsion, creates very similar spatio-temporal structures



	dive	g_swing	kick	lift	h_ride	run	skate	b_swing	h_swing	walk
dive	0.86			0.07	0.07					
g_swing		0.39	0.06		0.06	0.06	0.06			0.39
kick	0.05		0.45	0.05	0.1	0.2	0.05	0.05		0.05
lift				1						
h_ride			0.08		0.67	0.25				
run	0.08		0.23		0.15	0.46		0.08		
skate			0.17	0.17		0.08	0.08	0.08		0.42
b_swing			0.05			0.05	0.1	0.8		
h_swing	0.08								0.92	
walk	0.05		0.05		0.05	0.05				0.82

	dive	g_swing	kick	lift	h_ride	run	skate	b_swing	h_swing	walk
dive	1									
g_swing		0.89			0.06					0.06
kick			0.95	0.05						
lift				1						
h_ride			0.08		0.92					
run					0.08	0.92				
skate							1			
b_swing								1		
h_swing									0.92	0.08
walk										1

Figure 8: Confusion matrices from classification results on the UCF-Sports dataset using SM-1 and SM-2.

to *run*. Specifically regarding the *golf swing* and *kick* actions, confusion is generated because of the models's lack of imposing locality from the video to keep the overall simplicity. For example, in the *golf swing* videos, since no masks were used to track the subject of interest, the algorithm detected other subjects that are standing and walking in the same scene. While this problem could have been alleviated exploiting the available masks, the single additional sparse code in SM-2 addressed it, and such step is certainly significantly simpler than any possible tracker. In addition, most of the videos containing the *kick* action were preluded by a *walk* or *run* action by the subject of interest and/or the surrounding people, and the confusion is therefore reasonable (see Figure 7). Nevertheless, the failure of SM-1 seems to be purely a consequence of the very simple labeling procedure, since SM-SVM attained a significantly higher performance using the same proposed model, just a different back-end classifier (and again, the simpler SM-2 completely solved the mentioned problems). This shows that the discriminative information in **S** and **g** are sufficient for action classification, even in such challenging environments.

Table 4: Results for the UCF-Sports dataset.

Method	**Overall Accuracy (%)**
Le *et al.* [20]	86.5
Wang *et al.* [34]	88.2
Kovashka *et al.* [17]	87.5
SM-SVM	87.6
SM-1	66.3
SM-2	**96.0**

Figure 9: Sample frames from the YouTube dataset.

4.4 YouTube

The YouTube Dataset[7] [21] 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, 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 [21], 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. Following [35], the videos were re-sampled to half the original resolution. We also converted all frames to grayscale values. Figure 9 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 Figure 10 shows the confusion matrices corresponding to SM-1 and SM-2. We obtain overall classification rates of 87%, 80.29%, and 91.9% from SM-SVM, SM-1, and SM-2, respectively. First, comparing the performance of SM-1 for this dataset as with the UCF-Sports dataset, we notice improved performance. This may be a consequence of the distribution of the classes, and the structure of the individual videos, which reduces the effect of the spatial locality in the data extracted. For example, some videos corresponding to the *spike* action contain a high number of background clutter factors (e.g, crowd and teammates), but these do not perform any of the actions from the rest of the set.

The accuracy attained by SM-SVM is already 4.4% higher that 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

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

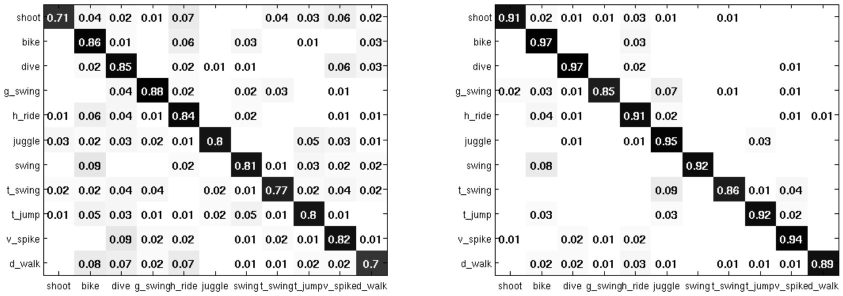


Figure 10: Confusion matrices from classification results on the YouTube dataset using SM-1 and SM-2.

the highest accuracy, as it decreased the scattered instances of misclassification obtained by SM-1 by implicitly imposing sparsity in a grouping fashion.

Table 5: Results for the YouTube dataset.

Method	**Overall Accuracy (%)**
Le *et al.* [20]	75.8
Wang *et al.* [34]	84.2
Ikizler-Cinbis *et al.* [14]	75.2
SM-SVM	88.18
SM-1	80.29
SM-2	**91.9**

4.5 Summary

Summarizing these results, we reported an increase in the classification accuracy of 2.6% in KTH, 2.8% in UT-Tower, 7.8% in UCF-Sports, and 7.7% 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 **A** to **S** and the local to global mapping do not degrade the discriminative information, but on the contraty, they enhance it.

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.

We are currently interested in incorporating locality to the model, which could provide additional insight for analyzing more sophisticated human interactions. We are also exploiting time-dependencies for activity-based summarization of motion imagery.

Acknowledgments: 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

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

[2] R. Blake and M. Shiffrar. Perception of human motion. *Annual Review of Psychology*, 58(1):47–73, 2007.

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

[4] C. Cadieu and B. A. Olshausen. Learning transformational invariants from natural movies. In *NIPS*, pages 209–216, 2008.

[5] A. Castrodad, Z. Xing, J. Greer, E. Bosch, L. Carin, and G. Sapiro. Learning discriminative sparse representations for modeling, source separation, and mapping of hyperspectral imagery (accepted for publication). *Geoscience and Remote Sensing, IEEE Transactions on*, 2011.

[6] A.S. Charles, B.A. Olshausen, and C.J. Rozell. Learning sparse codes for hyperspectral imagery. *IEEE Journal of Selected Topics in Signal Processing*, 2011.

[7] C.C. Chen, M. S. Ryoo, and J. K. Aggarwal. UT-Tower Dataset: Aerial View Activity Classification Challenge. http://cvrc.ece.utexas.edu/SDHA2010/Aerial_View_Activity.html, 2010.

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

[9] T. Dean, R. Washington, and G. Corrado. Recursive sparse, spatiotemporal coding. In *ISM*, pages 645–650, 2009.

[10] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2nd Joint IEEE International Workshop on*, pages 65–72, 2005.

[11] D. L. Donoho. High-dimensional data analysis: the curses and blessings of dimensionality. In *American Mathematical Society Conf. Math Challenges of the 21st Century*, 2000.

[12] J. Gall, A. Yao, N. Razavi, L. van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition (accepted for publication). *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2011.

[13] K. Guo, P. Ishwar, and J. Konrad. Action recognition using sparse representation on covariance manifolds of optical flow. In *AVSS*, pages 188–195, 2010.

[14] N. Ikizler-Cinbis and S. Sclaroff. Object, scene and actions: combining multiple features for human action recognition. In *ECCV*, pages 494–507, 2010.

[15] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, pages 1–8, 2007.

[16] A. Klser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008.

[17] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *CVPR*, pages 2046–2053, 2010.

[18] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, pages 432–439, 2003.

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

[20] Q.V. Le, W.Y. Zou, S.Y. Yeung, and A.Y. Ng. Learning hierarchical spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011.

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

[22] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, March 2010.

[23] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *NIPS*, pages 1033–1040, 2008.

[24] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *CVPR*, pages 3501–3508, 2010.

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

[26] M.S. Ryoo, C.C. Chen, J.K. Aggarwal, and A. R. Chowdhury. An overview of contest on semantic description of human activities (sdha) 2010. In *ICPR-Contests*, pages 270–285, 2010.

[27] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR*, pages 32–36, 2004.

[28] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ACM Multimedia*, pages 357–360, 2007.

[29] L. Shao and R. Mattivi. Feature detector and descriptor evaluation in human action recognition. In *CIVR*, pages 477–484, 2010.

[30] P. Sprechmann and G. Sapiro. Dictionary learning and sparse coding for unsupervised clustering. In *ICASSP*, 2010.

[31] G.W. Taylor, R. Fergus, Y.L. Le Cun, and C. Bregler. Convolutional learning of spatio-temporal features. In *ECCV*, pages VI: 140–153, 2010.

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

[33] R. Vezzani, B. Davide, and R. Cucchiara. HMM based action recognition with projection histogram features. In *ICPR*, pages 286–293, 2010.

[34] H. Wang, A. Kläser, C. Schmid, and L. Cheng-Lin. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176, Apr. 2011.

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

[36] G. Willems, T. Tuytelaars, and L. van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, pages II: 650–663, 2008.

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

[38] L. Yeffet and L. Wolf. Local trinary patterns for human action recognition. In *ICCV*, pages 492–497, 2009.