Learning Sparse Representations for Human Action Recognition

Tanaya Guha, *Student Member*, *IEEE*, and Rabab Kreidieh Ward, *Fellow*, *IEEE*

Abstract—This paper explores the effectiveness of sparse representations obtained by learning a set of overcomplete basis (dictionary) in the context of action recognition in videos. Although this work concentrates on recognizing human movements—physical actions as well as facial expressions—the proposed approach is fairly general and can be used to address other classification problems. In order to model human actions, three overcomplete dictionary learning frameworks are investigated. An overcomplete dictionary is constructed using a set of spatio-temporal descriptors (extracted from the video sequences) in such a way that each descriptor is represented by some linear combination of a small number of dictionary elements. This leads to a more compact and richer representation of the video sequences compared to the existing methods that involve clustering and vector quantization. For each framework, a novel classification algorithm is proposed. Additionally, this work also presents the idea of a new local spatio-temporal feature that is distinctive, scale invariant, and fast to compute. The proposed approach repeatedly achieves state-of-the-art results on several public data sets containing various physical actions and facial expressions.

Index Terms—Action recognition, dictionary learning, expression recognition, overcomplete, orthogonal matching pursuit, sparse representation, spatio-temporal descriptors.

1 Introduction

Sparse signal representation has emerged as an extremely successful tool for analyzing a large class of signals. Many signals like audio, images, video, etc., can be efficiently represented with linear superposition of only a small number of properly chosen basis functions. Although the use of orthogonal bases like Fourier or Wavelets is widespread, the latest trend is to use overcomplete basis—where the number of basis vectors is greater than the dimensionality of the input vector. A set of overcomplete basis (called a *dictionary*) can represent the essential information in a signal using a very small number of nonzero elements. This leads to higher sparsity in the transform domain as compared to that achieved by sinusoids or wavelets alone. Such compact representation of signals is desired in many applications involving efficient signal modeling.

With overcomplete basis however greater difficulties arise; because a full-rank dictionary matrix $\mathbf{\Phi} \in \mathbb{R}^{n \times m}$ $(n < m)$ creates an underdetermined system of linear equations $\mathbf{b} = \mathbf{\Phi x}$ having an infinite number of solutions. The goal is to find a *sparse* solution, i.e., $\mathbf{x} \in \mathbb{R}^n$ should contain no more than k $(k \ll n)$ nonzero elements. This in general is an NP hard problem. Nevertheless, in the last few years researchers have found practical and stable ways of solving such underdetermined systems via linear programming and greedy algorithms. For certain conditions like high sparsity (small k), small overcompleteness factor (m/n), etc., sparse representations are also shown to be stable in the presence of noise [1].

A crucial question is how to select the bases for the dictionary $\mathbf{\Phi}$. Predefined basis functions like curvelets, bandlets, variants of wavelets, etc., can be used. However, the success of such prespecified dictionaries is often limited by their suitability in capturing the structure in the signals under consideration. For example, image contents have sparse representation over wavelet dictionary, but audio signals are better represented by sinusoids. Another way of constructing a dictionary is to use training samples of the signal directly as the dictionary columns [2]. A more generalized approach is to *learn* the basis vectors that are specialized in representing the signal in question. Recent research shows that it is possible to learn a dictionary by fitting a set of overcomplete basis vectors to a collection of training samples [3], [4], [5]. Since each basis vector (atom) captures a significant amount of structure present in the given data [3], the learned dictionaries are more flexible than the predefined ones and can yield even more compact representation. Learned dictionaries have been shown to produce state-of-the-art results in image compression [5], color image restoration [6], and denoising [7].

This paper explores the usefulness of sparse representation obtained using learned dictionaries for video classification, looking particularly at the problem of recognizing human actions—both physical actions and facial expressions. Recognizing human actions is a challenging problem due to real-world conditions like partial occlusion, background clutter, changes in scale, viewpoint, and appearance. We propose to model human actions by learning overcomplete dictionary and its corresponding sparse representation using spatio-temporal descriptors extracted from the videos. Obeying the classical supervised learning paradigm, three different dictionary training frameworks

• *The authors are with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada. E-mail: {tanaya, rababw}@ece.ubc.ca.*

Manuscript received 3 Feb. 2011; revised 6 Nov. 2011; accepted 27 Nov. 2011; published online 19 Dec. 2011.
Recommended for acceptance by F. Mori.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2011-02-0077.
Digital Object Identifier no. 10.1109/TPAMI.2011.253.

are investigated: 1) *Shared*—one dictionary for all classes, 2) *Class-specific*—one dictionary per class, and 3) *Concatenated*—concatenation of the class-specific dictionaries. Pertaining to each framework a classification strategy is proposed. We consider two spatio-temporal features: the Cuboids descriptor [8] and a newly developed one which we call the Local Motion Pattern descriptor. We also show that the Random Projection (RP) can be used as a dimensionality reduction tool in the proposed framework which significantly reduces the computational cost. For a critical evaluation of our approach, various experiments were performed on the following public data sets: Weizmann Action [9], Weizmann Robustness [9], Ballet [10], UCF Sports [11], and Facial Expression data set [8]. These data sets pose various challenges in terms of real actions, complex motion, low-quality data, background clutter, partial occlusion, viewpoint, scale, and illumination changes. The proposed sparse representation-based approach repeatedly achieves state-of-the-art results indicating the efficacy of our system.

The rest of the paper is organized as follows: Section 2 discusses the previous work on sparse representation-based classification and on action recognition. Section 3 summarizes the contributions of our work. Section 4 describes the proposed approach in detail and Section 5 presents experimental results. We conclude the paper in Section 6, which discusses the overall effectiveness and limitations of the proposed approach and also suggests possible directions to future work.

2 Related Work

The theory of sparse representation aims at finding efficient and compact representations for signals and is primarily suitable for problems like denoising, compression, inpainting, etc. Recently, a work on image-based face recognition [2] showed that sparse representation is naturally discriminative; it selects only those basis vectors among many that most compactly represent a signal and therefore is also useful for classification. In [2], a single overcomplete dictionary is formed by concatenating the vectorized training samples of all classes. Given a test image, its sparsest representation over the dictionary is found by ℓ_1 minimization. The underlying assumption of this method is that a good number of training samples are available per class and they span the sample space well.

In [12] and [13], texture is modeled by learning dictionaries from raw image patches. The dictionaries are used for texture synthesis and segmentation. In [12], the dictionaries are learned using K-SVD algorithm [5]. In [13], the dictionaries are built by jointly optimizing an energy formula containing both sparse reconstruction and class discrimination components. The object recognition approach presented in [14] moved from pixel domain to feature domain by obtaining sparse decomposition of the Scale Invariant Feature Transform (SIFT) features [24]. The authors replace vector quantization by sparse coding in order to learn a single codebook and stick to traditional classification methods. High recognition accuracy is achieved in [14] using a spatial pyramid max-pooling scheme and linear Support Vector Machine (SVM) classifier. The problem of action recognition is addressed in [15] in a similar manner. Instead of pyramid max-pooling, however, [15] uses single scale max-pooling which ignores most of the sparse coefficients and thus does not fully exploit the strength of sparse representations.

Modeling an action in a video sequence starts with a powerful video representation. A popular approach is to describe an action sequence with some kind of motion descriptors [8], [16], [17], [18]. Motion descriptors capture the important spatio-temporal patterns that characterize a particular action as well as discriminate it from others. In [8], a separable filter-based feature detector and a variety of descriptors are proposed. A dense sampling method that extracts video patches at regular position and scale is proposed for object recognition in [19]. Motivated by this work, the authors of [20] show that dense sampling can also handle difficult action recognition tasks quite well. For a detailed evaluation of different motion descriptors the reader may refer to [20].

After the motion features are computed, a certain action is often represented as a collection of codewords in a predefined codebook. This is the well-known Bag-of-Words (BoW) model, which has been adopted by many computer vision researchers [8], [21], [22]. In its basic form, this modeling approach disregards all spatial and temporal relationships among the codewords. An unsupervised learning approach that uses BoW representation for human action recognition is presented in [21]. Other approaches to analyze human actions include treating an action sequence as 2D templates like Motion Energy Image (MEI), Motion History Image (MHI) [23], and as 3D space-time shape volumes [9].

2.1 BoW Modeling and Dictionary Learning

The relationship between the codebook used in BoW modeling and the dictionary learned for sparse representation is particularly interesting. A codebook and a dictionary are conceptually similar in the sense that they both consist of a set of representative vectors learned from a large number of samples. These representative vectors are called codewords in the context of BoW modeling and atoms otherwise. In BoW modeling, a codebook is learned by clustering, using vector quantization. During clustering, each sample vector is assigned to the codeword that is closest to it in terms of euclidean distance. This can be interpreted as the extreme sparsity constraint, where each sample vector is allowed to be approximated by one and only one codeword. This leads to a considerable amount of approximation error. Also, the codebook size (number of codewords) has to be increased as the data exhibit more and more variation. To reduce the approximation error and create compact dictionaries, the sparsity constraint can be relaxed by allowing a few more codewords to participate in the approximation process. This coincides with the idea of sparse representation-based dictionary learning, which is a recent development in theoretical signal processing. In dictionary learning, each sample vector is approximated by a weighted sum of a small number of dictionary atoms. Thus, learning overcomplete dictionaries can be considered as a generalization of the vector quantization-based codebook learning process in BoW modeling approach.

3 Contributions

Prior work on classification using sparse representation has mainly dealt with images. But videos, being functions of space and time, pose a bigger challenge. The main contributions of our work are summarized below:

- Our primary objective is to explore the applicability of sparse representation obtained using learned dictionaries for classification, looking particularly at the problem of human action recognition. Currently, our approach works for a wide range of motions—both body movements and facial expressions. The proposed approach is fairly general and can also be used to address other classification tasks.
- The dictionaries are learned by sparse coding so as to obtain richer and more compact representation of the action sequences compared to the vector quantization-based ones. Three options for constructing the dictionaries are investigated: shared, class-specific, and concatenated. Although building a shared dictionary is a familiar concept in action recognition, the other two are new. Novel classification algorithms are also developed for each dictionary type.
- A simple yet effective method for detecting and computing important motion patterns is proposed. The features are designed to capture the distinctive, local, space-time motion patterns that are very fast to compute. They are named the Local Motion Pattern (LMP) descriptors.
- Our work is one of the few to use RP in a classification framework. In the proposed approach, RP successfully reduces the computational cost in two ways—by avoiding the cost of traditional methods like Principal Component Analysis (PCA), and by limiting the dimension of input samples and thereby the dimension of required overcomplete dictionary.

4 The Proposed Approach

Our approach broadly consists of four stages: computation of the spatio-temporal motion descriptors, dimensionality reduction of the descriptors, learning overcomplete dictionaries using the lower-dimensional descriptors, and classification using the dictionaries and/or the corresponding sparse representations. Below, we describe each stage in detail.

4.1 Computation of Spatio-Temporal Features

The first stage is to develop a rich spatio-temporal representation for each action sequence. To obtain such a representation we choose the Cuboid [8] descriptors (since it is widely popular and generates a good number of features) and additionally design a new descriptor called the LMP descriptors. Note that our approach is not dependent on any particular spatio-temporal feature as long as it generates a good number of features. Features that are too few are not desirable for learning a good dictionary.

Motivation behind the new motion descriptor: To compute spatio-temporal features, typically some response function has to be computed at every location in a video where the extrema points correspond to the keypoints. This detection part is responsible for most of the computational load. It has been shown in [19] and [20] that even if such a detection process is omitted and patches are extracted at regular intervals from images/videos, the resulting features can produce highly accurate results. This is known as dense sampling. The bottleneck here is that the number of features required for dense sampling is 15-20 times greater than that needed for traditional feature detectors [20]. In this section, we design a feature detector that resolves the issues of both dense sampling and space-time feature detection. It significantly lowers the computational load by detecting the keypoints in spatial domain only, but at the same time retains the important temporal information. Also, it does not generate an inconveniently large number of features. The LMP descriptor is a fast and simple alternative to dense sampling and traditional feature detectors.

4.1.1 Cuboids Features

The Cuboid detector relies on separable linear filters for computing the response function of a video sequence $\mathbf{V}(x, y, t)$. The response function is of the form $R = (\mathbf{V} * g * h_{ev})^2 + (\mathbf{V} * g * h_{od})^2$, where $g(x, y; \sigma)$ is the 2D Gaussian smoothing function (applied only in the spatial domain) and h_{ev} and h_{od} are a quadrature pair of 1D Gabor filters (applied in the temporal direction). The 1D Gabor filters are defined as $h_{ev} = -cos(2\pi t\omega)\exp^{-t^2/\tau^2}$ and $h_{od} = -sin(2\pi t\omega)\exp^{-t^2/\tau^2}$, where $\omega = 4/\tau$. The parameters σ and τ roughly correspond to the spatial and temporal scales. Keypoints are detected at the local maxima of the response function. The video patches extracted at each of the keypoints are converted to a descriptor. A number of ways to compute descriptors from video patches have been suggested in [8]. Among those, gradient-based descriptors like Histogram of Gradients (HoG) and concatenated gradient vectors are the most reliable ones [8]. For more details about the Cuboid features please refer to [8].

4.1.2 Proposed LMP Features

Feature detection: We define a *local motion pattern* as a distinctive, scale-invariant region that contains significant information about the local variations of the signal along both spatial and temporal dimensions. It was noted in [8] that the extrema points are often located at the regions having spatially distinguishing structure. Consequently, we deduce that the local motion patterns should correspond to the temporal variations in such spatially distinctive regions over a short period of time. Our purpose is to detect the spatially distinctive points and then capture the temporal changes in the neighborhood of those points.

Consider a video sequence $\mathbf{V}(x, y, t)$ consisting of f frames. It is first partitioned into S segments: $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_S]$ (as shown in Fig. 1) such that each segment contains $l = f/S$ consecutive frames. The number of frames in a segment, l, corresponds to the temporal resolution at which $\mathbf{V}$ is analyzed. The smaller the value of l the finer is the resolution. At any given resolution l is required to be large enough to accommodate small movements of the subject but not too large to have any major changes.

In order to extract spatially distinguishing structures we employ a 2D keypoint detector and locate keypoints at the first frame of every temporal segment. Say, ρ keypoints are detected in the first frame of a segment $\mathbf{V}_i$. We are interested in observing how the temporal information around each of

Fig. 1. Multiple temporal scales analysis of a video sequence partitioned into four and eight temporal segments for computation of the LMP descriptors.

these ρ keypoints changes over the remaining $(l-1)$ frames. This can be handled by prealigning the subjects (when translation is involved) in all the frames of $\mathbf{V}_i$ w.r.t. a reference point. Then, fixing the coordinate values obtained for the keypoints in the first frame, small video patches of dimension $(\eta \times \eta \times l)$ are extracted around each of the ρ key points, in every $\mathbf{V}_i$, $i = 1, 2, ..S$.

The prealignment of frames simplifies the process of patch extraction. Often, such prealigned sequences are the output of the tracking procedures used to detect the subject of interest. However, it requires a good bounding box and may be difficult in the cases of background clutter or partial occlusion. An alternative to prealignment of the figures is to find the points corresponding to the keypoints detected in the first frame in the next frames, for example, by SIFT feature matching [24]. Note that prealignment removes all information about a subject's translation, but translation does not contribute much to the recognition process anyway. This prealignment step is also adopted in [9] and [18].

The descriptor: Every keypoint is associated with a spatio-temporal cube of size $(\eta \times \eta \times l)$. Each cube captures the local space-time changes of the signal and represents a significant motion pattern. The spatio-temporal cubes are extracted in all temporal segments of $\mathbf{V}$. In order to obtain a robust descriptor for each spatio-temporal cube, we first perform 2D Gaussian blurring of each cube in the spatial domain so as to ignore minor variations. This increases the robustness of the descriptor against noise and positional uncertainties that are likely to occur from imperfect segmentation or improper alignment, if performed. But the cubes should not be smoothed along the temporal direction so as not to ruin the small temporal variations we are particularly interested in.

Let us denote a blurred cube as $\mathbf{v} \in \mathbb{R}^{\eta \times \eta \times l}$, which is basically a series of l small patches. After removing the mean of $\mathbf{v}$, the second (variance, $\mathbf{M}_2$), third (skewness, $\mathbf{M}_3$), and fourth (kurtosis, $\mathbf{M}_4$) central moments are computed for each pixel along the temporal direction. We define the moment matrix $\mathbf{M}_r$, $r = \{2, 3, 4\}$, associated with $\mathbf{v}$ as follows:

$$\mathbf{M}_r = \left[m_{ij}\right] \quad i, j = 1, 2, \ldots \eta, \tag{1}$$

where

$$m_{ij} = \frac{1}{l}\sum_{t=1}^{l}\left(v_{ijt}\right)^r. \tag{2}$$

Here, v_{ijt} is the pixel value at location $\{i, j\}$ of the tth patch. Each moment matrix $\mathbf{M}_r$, $r = \{2, 3, 4\}$, is transformed to a vector $\mathbf{m}_r \in \mathbb{R}^{\eta^2}$. The three moment vectors corresponding

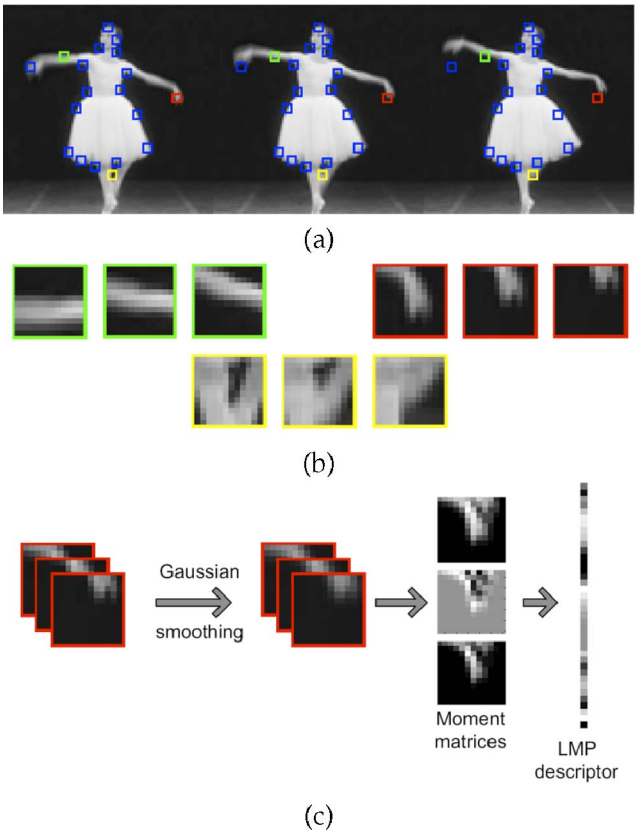


Fig. 2. (a) A temporal segment consisting of three consecutive video frames. The 2D keypoints are identified in the first frame using improved Harris keypoint detector. The positions of the same keypoints are shown in the next two frames. (b) Patches are extracted around each keypoint at each frame. Three space-time cubes associated with the three keypoints (green, red, yellow) are shown. Each cube contains patches extracted from the three frames. (c) Conversion of a cube to an LMP descriptor: Gaussian blurring of the cube is followed by the computation of the second, third, and fourth central moments in the temporal dimension and transformation of the three moment matrices into one vector. (This image is best viewed in color.)

to three values of r are concatenated on top of each other to form a single vector $\mathbf{m} \in \mathbb{R}^d$ where $d = 3\eta^2$:

$$\mathbf{m} = \begin{bmatrix} \mathbf{m}_2 \\ \mathbf{m}_3 \\ \mathbf{m}_4 \end{bmatrix}. \tag{3}$$

The vector $\mathbf{m}$ is an LMP descriptor. A number of such descriptors that collectively characterize a human action is extracted from each video sequence. The process of computing the LMP descriptors is illustrated in Fig. 2. The advantages of these proposed descriptors are as follows:

- *Computational efficiency*—Assume that the video frames are prealigned. The order of computational complexity of detecting keypoints in an image, using for example, the Harris interest point detector, is $\mathcal{O}(n)$, where n is the number of pixels in the image. For a video sequence divided into S number of temporal segments, keypoints have to be detected only in S number of images. If we consider T temporal scales $(T \geq 1)$, the complexity is $\mathcal{O}(nC) \sim \mathcal{O}(n)$, where $C = \sum_{j=1}^{T} S_j$ is a small constant and S_j is the number of temporal segments at scale j. Thus, the order of

TABLE 1
Quantitative Comparison between Cuboids and LMP

	Cuboids	LMP
video size	101×101 × 84	101×101 × 84
temporal scales	3	3
spatial scale	2	2
features extracted	438	474
run time (sec)	16.70	5.08

complexity of extracting the spatio-temporal cubes is equal to that of the 2D keypoint detector being used. Evidently the complexity of 2D extrema detection is much lower than the 3D extrema detection used to find the 3D spatio-temporal keypoints in [8], [16], [17]. From Table 1, we can see that LMP is almost three times as fast as the cuboids.

- *Flexibility*—One can choose from a large pool of 2D keypoint detectors based on the application and data-type. Descriptors can be computed for a variety of data types such as silhouettes, blobs, and plain grayscale images. Background subtraction is not necessary.
- *Scale invariance*—Temporal and spatial scale invariance is easy to achieve by using a multiscale 2D keypoint detector and multiple temporal resolutions.

The demerit of this feature extraction method is the cost of prealignment of the video frames or alternatively, tracking the keypoints in the consecutive frames.

4.2 Random Projections

Motion descriptors are typically high-dimensional. A Cuboid-HoG descriptor is of dimension $[1440 \times 1]$ and an LMP feature vector for a patch of size (24×24) is of dimension $[1728 \times 1]$. Recall that $\mathbf{\Phi} \in \mathbb{R}^{n \times m}$, where n is the descriptor dimension and usually $m \geq 2n$. The features, if used with the original dimension, ask for more than 2,500 dictionary atoms to be learned in order to secure a sparse representation. This high dimensionality seriously limits the speed and practical applicability of our approach. A natural solution is to reduce the dimensionality. The application of standard methods like PCA, Linear Discriminant Analysis (LDA), etc., to obtain lower dimensional representation is well-known. Recently, Random Projection [25] has emerged as a powerful tool in dimensionality reduction. Theoretical results show that the projections on a random lower dimensional subspace can preserve the distances between vectors quite reliably. The advantages of RP are that it is data independent, simple, and fast.

Consider a set of p descriptors obtained from a video sequence, where each descriptor is of length d. This set can be represented as a matrix $\mathbf{D} \in \mathbb{R}^{d \times p}$. The original d-dimensional descriptors are projected onto an n-dimensional subspace ($n \ll d$) by premultiplying the descriptor matrix $\mathbf{D}$ by a random matrix $\mathbf{R} \in \mathbb{R}^{n \times d}$. In practice, any normally distributed $\mathbf{R}$ with zero mean and unit variance serves the purpose. There exist other choices of non-Gaussian random matrices that can save on computations even more. The dimensionality reduction step then simplifies to a simple matrix multiplication, given by

$$\mathbf{Y} = \mathbf{R}\mathbf{D}, \tag{4}$$

where the reduced data matrix $\mathbf{Y} \in \mathbb{R}^{n \times p}$ contains projections (not true projections because the vectors are not orthogonal) of $\mathbf{D}$ on some random n-dimensional subspace.

4.3 Dictionary Learning

The next stage is to learn the overcomplete dictionaries and the corresponding sparse representations using the motion descriptors. We start with briefly describing the dictionary learning algorithm.

Consider a set of lower-dimensional descriptors $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{p}$, $\mathbf{y}_i \in \mathbb{R}^n$. We wish to learn a dictionary $\mathbf{\Phi} \in \mathbb{R}^{n \times m}$ $(m > n)$ over which $\mathbf{Y}$ has a sparse representation $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{p}$, $\mathbf{x}_i \in \mathbb{R}^m$, such that each $\mathbf{x}_i$ contains k ($k \ll n$) or fewer nonzero elements. This is formally written as the following optimization problem:

$$\min_{\mathbf{\Phi},\mathbf{X}} \left\{ \left\| \mathbf{Y} - \mathbf{\Phi}\mathbf{X} \right\|_F^2 \right\} \text{ subject to } \|\mathbf{x}\|_0 \leq k, \tag{5}$$

where $\|.\|_F$ is the Frobenius norm and $\|.\|_0$ is the ℓ_0 seminorm that counts the number of nonzero elements in a vector. To solve (5), a recently developed dictionary learning algorithm, known as K-SVD [5], is used. K-SVD iteratively solves (5) by performing two steps at every iteration: 1) sparse coding and 2) dictionary update. In the sparse coding step, $\mathbf{\Phi}$ is kept fixed and $\mathbf{X}$ is computed.

$$\min_{\mathbf{X}} \left\{ \left\| \mathbf{Y} - \mathbf{\Phi}\mathbf{X} \right\|_F^2 \right\} \text{ subject to } \|\mathbf{x}\|_0 \leq k. \tag{6}$$

Note that the expression in (6) also has an ℓ_0 term. Alhough ℓ_0 provides a straightforward notion of sparsity, it makes the problem nonconvex. Solving (6) accurately is an NP-hard problem. Nevertheless, approximate solutions are provided by greedy algorithms like Matching Pursuit (MP), Orthogonal Matching Pursuit (OMP), and by more sophisticated approaches such as Basis Pursuit (BP). BP replaces the ℓ_0 term with an ℓ_1 penalty so as to transform the problem to a convex one. Some other solvers also suggest the use of the ℓ_p norm, $p \leq 1$, as a replacement to the ℓ_0 norm. In this work, we have used OMP to solve (6), as in the original K-SVD paper [5], because it is fast, easy to implement, and fairly accurate [5]. For the same reasons, we have used OMP to solve all the sparse approximation problems presented in this paper.

In the dictionary update stage, the atoms of dictionary $\mathbf{\Phi}$ are updated sequentially, allowing the relevant coefficients in $\mathbf{X}$ to change as well. Updating an atom in $\mathbf{\Phi}$ involves computing a rank-one approximation of a residual matrix:

$$\mathbf{E}_i = \mathbf{Y} - \widetilde{\mathbf{\Phi}_i}\widetilde{\mathbf{X}_i}, \tag{7}$$

where $\widetilde{\mathbf{\Phi}_i}$ and $\widetilde{\mathbf{X}_i}$ are formed by removing the ith column from $\mathbf{\Phi}$ and the ith row from $\mathbf{X}$. This rank-one approximation is computed by subjecting $\mathbf{E}_i$ to a Singular Value Decomposition (SVD). For a detailed description of the K-SVD algorithm please refer to [5].

Assume that there are K classes. For each class, a set of motion descriptors is extracted from each of the training sequences. In order to model these descriptors using learned dictionaries, we consider three options:

- *Shared*—Learning a single dictionary for all classes.

- *Class-specific* dictionaries—Learning K dictionaries, one for each class.
- *Concatenated* dictionaries—A single dictionary formed by concatenating K class-specific dictionaries.

4.4 Shared Dictionary

In this framework a single, shared dictionary $\mathbf{\Phi}$ is learned for all K classes so that multiple classes can share some common dictionary elements. Since the dictionary is learned only once it saves some computation, though a bigger dictionary is needed to accommodate the variations of all classes. The learning process has to be repeated whenever a new class is added to the system.

The matrix $\mathbf{Y}$ contains the descriptors obtained from the training samples of all classes and $\mathbf{X}$ contains their corresponding sparse representations over the learned shared dictionary $\mathbf{\Phi}$ (refer to (5)). The sparse coefficients in a column vector $\mathbf{x}_i \in \mathbf{X}$ present the contribution of all the dictionary atoms in approximating the descriptor $\mathbf{y}_i \in \mathbf{Y}$. The sparse coefficients associated with all the descriptors of a particular class thus collectively demonstrate the contribution of the dictionary atoms toward the representation of that class. Hence, some statistics of these sparse coefficients (sometimes called descriptors codes), if computed, will be able to characterize that class. A popular statistical representation is the coefficient histogram. Let the ith class have a sparse decomposition $\mathbf{X}_i = \{\mathbf{x}_j\}_{j=1}^{p}$ over $\mathbf{\Phi}$. Then its coefficient histogram $\mathbf{h}_i$ is computed as follows:

$$\mathbf{h}_i = \frac{1}{p}\sum_{j=1}^{p} \mathbf{x}_j. \tag{8}$$

Another popular alternative is to compute histograms for individual training sample and train a multiclass SVM classifier with them. We explore both the alternatives in the experiments section.

Given a query video sequence $\mathbf{V}_Q$, it is represented by a set of motion descriptors $\mathbf{Q} = \{\mathbf{q}_j\}_{j=1}^{q}$, $\mathbf{q}_j \in \mathbb{R}^n$. The recognition algorithm that uses class histograms is described below. The SVM-based classification method is well-known and therefore not described here.

Classification using the shared dictionary

- Learn a single shared dictionary $\mathbf{\Phi}$ as in (5)
- Compute the coefficient histograms $\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_K$, one for each class by (8)
- Given $\mathbf{Q}$, find its sparse representation $\mathbf{X}_Q$ over $\mathbf{\Phi}$

$$\min_{\mathbf{X}_Q} \{\|\mathbf{Q} - \mathbf{\Phi}\mathbf{X}_\mathbf{Q}\|_F^2\} \text{ s.t. } \|\mathbf{x}_Q\|_0 \leq k_1$$

- Compute the histogram $\mathbf{h}_Q$ pertaining to $\mathbf{Q}$
- Estimated class $= \text{argmax}_{i \in 1,2,\ldots,K}\ \mathbf{h}_Q^T \mathbf{h}_i$

4.5 Class-Specific Dictionaries

This framework learns K dictionaries $\mathbf{\Phi}_1, \mathbf{\Phi}_2, \ldots, \mathbf{\Phi}_K$, one for each class. One advantage of having class-specific dictionaries is that each class is modeled independently of the others and hence the painful repetition of the training process when a new class of data is added to the system is no longer necessary. This also indicates the possibility of parallel implementation.

The idea is that a dictionary tailored to represent one particular action will have an *efficient* representation for this class and at the same time will be *less efficient* in representing actions belonging to a different class. The *efficiency* here refers to the lower reconstruction error, while sparsity is constant. We exploit this inherent discriminative nature of the class-specific dictionaries and develop an efficient classification technique that we call the Random Sample Reconstruction.

Random sample reconstruction (RSR): Recall that the query video sequence $\mathbf{V}_Q$ is represented by a collection of descriptors as $\mathbf{Q} = \{\mathbf{q}_j\}_{j=1}^{q}$, $\mathbf{q}_j \in \mathbb{R}^n$. A simple way to classify $\mathbf{V}_Q$ is to find the K approximations of $\mathbf{Q}$ given by each of the K learned dictionaries and their corresponding reconstruction errors e_i, $i = 1, 2, \ldots K$, i.e.,

$$e_i = \|\mathbf{Q} - \mathbf{\Phi}_i \hat{\mathbf{X}}_{Q_i}\|_2^2, \tag{10}$$

where

$$\hat{\mathbf{X}}_{Q_i} = \underset{\mathbf{X}_Q}{\text{argmin}} \left\|\mathbf{Q} - \mathbf{\Phi}_i \mathbf{X}_{Q_i}\right\|_F^2 \text{ s.t. } \left\|\mathbf{x}\right\|_0 \leq k_2, \tag{11}$$

$\mathbf{X}_{Q_i} = \{\mathbf{x}_j\}_{j=1}^{q}$, $\mathbf{x}_j \in \mathbb{R}^m$, is the sparse representation of $\mathbf{Q}$ over $\mathbf{\Phi}_i$, $i = 1, 2, \ldots, K$. Then the estimated class of $\mathbf{V}_Q$ is the class that yields the smallest e_i:

$$\hat{i}_Q = \underset{i \in [1, 2 \ldots K]}{\text{argmin}}\, e_i. \tag{12}$$

This method discriminates on the basis of reconstruction error, which has been proven to be quite useful for texture classification [12], [13]. We will refer to this method as the *Simple Reconstruction* method.

In a complex problem like action recognition, a strong presence of outliers in $\mathbf{Q}$ is highly probable due to the errors in keypoint detection, noisy data, occlusion, etc. In the presence of high percentage of outliers, if all the descriptors in $\mathbf{Q}$ are used for reconstruction, the resulting reconstruction error will hardly be a reliable means of classification. In order to build a robust classifier, we propose the idea of Random Sample Reconstruction. This is motivated by the celebrated RANdom SAmple Concensus (RANSAC) algorithm [26]. RANSAC finds part of the data that best fits a given model, whereas RSR solves an even more difficult problem—finding both the best model (among a number of probable ones) and the part of the data that best fits the chosen model.

The basic assumption of the proposed RSR algorithm is that the best model and its coefficients can be estimated by a small number of good data points, i.e., error-free descriptors. Let the number of good data points (a subset of the available data points) be s, $s \ll q$, where q is the number of all available data points (number of query descriptors). Let the probability of selecting a good data point be ω and the probability of observing an outlier is $(1 - \omega)$. If we perform Λ trials and in each trial select s random data points, the probability of selecting at least one error-free set of s data points is $1 - (1 - \omega^s)^{\Lambda}$. We want to ensure that such a set can be selected with a probability P:

$$1 - (1 - \omega^s)^{\Lambda} = P. \tag{13}$$

For a given value of P and ω, the value of Λ that ensures the success of selecting an error-free data set is computed as

$$\Lambda = \frac{\log(1-P)}{\log(1-\omega^s)}. \quad (14)$$

At every trial, a random subset of s descriptors is selected. Let this working subset be denoted as $\mathbf{Q}_s$. The best model (dictionary) for $\mathbf{Q}_s$ is estimated by the simple reconstruction method. The descriptors that are not in $\mathbf{Q}_s$ are then approximated by the estimated model. The descriptors for which the reconstruction error is below a certain threshold are called the inliers. Our algorithm eventually selects the model that has the largest number of inliers. Note that s, the number of good data points is unknown. So, for our experiments, s is set to 1 percent of the total number of data points, i.e., $s = 0.01q$. The values of ω and Λ are updated at each iteration. The proposed algorithm is nondeterministic, i.e., it can determine the class only with a certain probability P. A less conservative value of P can be used to achieve faster convergence. The full description of the proposed RSR algorithm is given below.

Random Sample Reconstruction (RSR)

Initialize:

- no. of inliers $I_0 = 0$.
- total no. of datapoints $= q$
- no. of good datapoints $= s$ such that $s \ll q$ (e.g. $s = 0.01q$)
- set a high probability value $P = 0.99$.

Compute: $\omega = \frac{s+I_0}{q}$ and $\Lambda = \frac{\log(1-P)}{\log(1-\omega^s)}$

Loop until $\Lambda = 0$

- Choose s random descriptors from $\mathbf{Q}$ to form $\mathbf{Q}_s \subset \mathbf{Q}$.
- Estimate the class of $\mathbf{Q}_s$ by (10) - (12).
- Let the estimated class of $\mathbf{Q}_s$ be $\hat{i}_s$ and the corresponding dictionary be $\mathbf{\Phi}_s$.
- For every $\mathbf{q}_i \notin \mathbf{Q}_s$
  - $\epsilon_i = \|\mathbf{q}_i - \mathbf{\Phi}_s \mathbf{x}_{s_i}\|_2$, where $\mathbf{x}_{s_i}$ is the sparse representation of $\mathbf{q}_i$ over $\mathbf{\Phi}_s$ with sparsity k_2
  - Count inliers: $I \leftarrow \{i : \epsilon_i \leq T_h\}$, $T_h =$ threshold.
- *Update:* If $|I| > I_0$
  - $I_0 \leftarrow |I|$
  - Estimated class $\leftarrow \hat{i}_s$
  - update ω and Λ

4.6 Concatenated Dictionary

The third option to construct a dictionary is by concatenating the class-specific dictionaries. A bigger dictionary $\mathbf{\Phi}_C$ is formed by concatenating K dictionaries together. Let us assume that originally the query sequence belongs to the class α. If $\mathbf{Q}$ is approximated by $\mathbf{\Phi}_C$, ideally, every $\mathbf{q} \in \mathbf{Q}$ should use only the atoms of $\mathbf{\Phi}_\alpha$ for its representation. Although this condition is difficult to achieve in practice (due to errors in $\mathbf{Q}$ and correlation among the class-specific dictionaries), we can still expect that the atoms of $\mathbf{\Phi}_\alpha$ should be used more than any other dictionary atoms. This results into a higher concentration of nonzero elements in the coefficients corresponding to $\mathbf{\Phi}_\alpha$. The classification algorithm is as follows:

Classification using the concatenated dictionary

- Form $\mathbf{\Phi}_C = [\mathbf{\Phi}_1 | \mathbf{\Phi}_2 | \dots | \mathbf{\Phi}_K]$
- Find $\mathbf{X}_Q$ by OMP
  $\min_{\mathbf{X}_Q} \|\mathbf{Q} - \mathbf{\Phi}_C \mathbf{X}_Q\|_2^2$ s.t. $\|\mathbf{x}\|_0 \leq k_3$
- $\mathbf{X}_Q$ is written as
  $\mathbf{X}_Q = [\mathbf{X}_{\mathbf{\Phi}1} | \mathbf{X}_{\mathbf{\Phi}2} | \dots | \mathbf{X}_{\mathbf{\Phi}K}]$
  where $\mathbf{X}_{\mathbf{\Phi}i}$ is the coefficient matrix corresponding to $\mathbf{\Phi}_i$.
- Estimated class $= \text{argmax}_{i \in 1,2,\dots,C} \|\mathbf{X}_{\mathbf{\Phi}i}\|_0$

Clearly, $\mathbf{Q}$ is block sparse; this is because the nonzero coefficients in $\mathbf{X}_Q$ occur in clusters. This encourages us to exploit block sparsity as an additional structure. But, each block in $\mathbf{\Phi}_C$ is an overcomplete dictionary, which makes it difficult to use block sparsity promoting algorithms like Block OMP (BOMP) [27]. We have used BOMP and observed that the experimental results are neither consistent nor very accurate.

5 Performance Evaluation

A critical experimental evaluation of the proposed approach is presented in this section. Our main objective is to evaluate the strength of the proposed sparse modeling and classification algorithms; a secondary goal is to test the effectiveness of the LMP descriptors. Evaluation is done on the basis of discriminating power, robustness against occlusion, viewpoint changes, variations in scale (spatial and temporal), illumination changes, and ability to model complex motions. We have used four public data sets that exhibit various motions in different conditions—starting from everyday actions to professional sports, complex ballet movements, and even facial expressions.

5.1 Parameter Settings

For feature extraction, both Cuboids and LMP descriptors use two spatial and three temporal scales. Descriptor parameters have been set such that they can use similar spatial and temporal resolutions. The Cuboid detector uses the following parameter values: $\sigma = [2, 4]$ and $\tau = [1, 1.2, 1.4]$. Among the different descriptors proposed in [8], we choose the gradient-based HoG descriptor since it often produces reliable results [8], [20]. Each Cuboid-HoG descriptor is of dimension 1,440. For the LMP descriptor, the spatially distinctive points are computed by an improved version of the Harris keypoint detector [28]. This 2D keypoint detector has been shown to outperform other keypoint detectors in terms of distinctiveness and stability of the detected points under various image transformations like rotation, illumination variation, scale, and viewpoint changes [28]. Any other 2D keypoint detector should work as well (e.g., Laplacian of Gaussian worked well for our experiments and produced similar results). The video patches are extracted at two spatial scales using the original frame size and a 1.5 times downsampled version of the same with $\eta \times \eta = [24 \times 24]$. Each video is analyzed at three temporal resolutions $S = 8, 10, 12$. An LMP descriptor is of dimension $3 * \eta^2 = 1{,}728$. The high-dimensional Cuboid/LMP descriptors are then projected on a random 128D space.

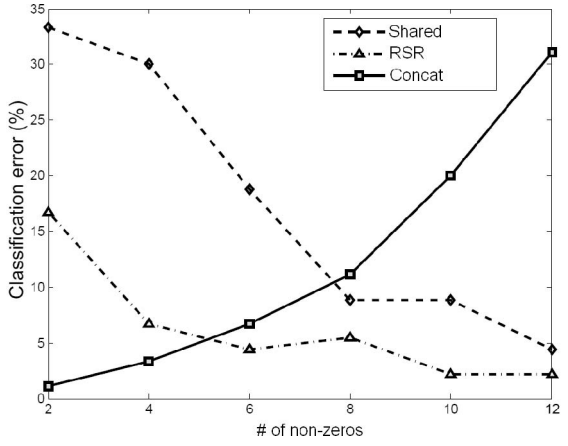


Fig. 3. Performance of the proposed classification algorithms with sparsity (k_i, $i = 1, 2, 3$) on the Weizmann action data set.

The random projection matrix $\mathbf{R}$ is constructed at every run during cross validation.

The shared dictionary $\mathbf{\Phi} \in \mathbb{R}^{128\times 512}$ and the class-specific dictionaries $\mathbf{\Phi}_i \in \mathbb{R}^{128\times 256}$, $i = 1, 2, \ldots, K$, are learned using $k = 12$ (approximately 10 percent of the dimension of the descriptor) and 20 K-SVD iterations. Experiments are performed separately for both of the descriptors under four settings:

1. shared dictionary with histogram correlation (Shared-Hist),
2. shared dictionary with linear SVM (Shared-SVM),
3. class-specific dictionary with RSR (RSR), and
4. concatenated dictionary (Concat).

There is a number of parameters to be chosen carefully. The straightforward way is to use cross validation. For this work, parameter sweep is done for the Weizmann action data set only and the same values are used for all other data sets. Note that, increasing patch size, number of scales or feature dimension improves recognition accuracy, but it also raises the computational load significantly. Also, the theory of sparse representation and dictionary learning is in a developing stage; how to set the parameters like optimal dictionary size, sparsity, etc., are some of the open issues.

In order to understand the impact of sparsity on the decision making process, we have run the recognition experiments on the Weizmann action data set for different values of k_1, k_2, and k_3. Fig. 3 shows that for the shared and RSR methods recognition accuracy increases as the number of nonzeros increases (i.e., sparsity decreases). On the other hand, in the case of concatenated dictionary, accuracy decreases with sparsity. This observation about a dictionary containing features from all classes that "the sparser the solution, the more accurate is the classification" agrees with that in [2]. The RSR algorithm provides lower recognition error over the varying range of sparsity, as it carefully selects the error-free (or less faulty) descriptors. The threshold parameter T_h required for the RSR algorithm is obtained empirically (e.g., 0.4*the maximum error). The leave-one-out strategy is adopted for the evaluation of all the data sets unless mentioned otherwise.

Fig. 4. Sample frames from the Weizmann action data set: bend (w1), jumping jack (w2), jump forward (w3), jump in place (w4), run (w5), gallop sideways (w6), skip (w7), walk (w8), wave one hand (w9), and wave both hands (w10).

5.2 Weizmann Action and Robustness Data Set

This benchmark data set, frequently used by researchers, provides a good platform for comparing the proposed approach with varied action recognition approaches under similar experimental setup. It consists of 90 low-resolution (180×144, deinterlaced 50 fps) video sequences of nine subjects, each performing 10 natural actions: bend, jumping jack, jump forward, jump in place, run, gallop sideways, skip, walk, wave one hand, and wave both hands. The database uses a fixed camera setting and a simple background. No occlusion or viewpoint changes are present originally. Variations in spatial and temporal scale are also minimal. Sample frames from this database are presented in Fig. 4. We have used the prealigned, background subtracted silhouettes provided by the authors of [9] *only* for this data set. The silhouettes are used to establish the versatility of the proposed LMP descriptor.

The performances of Cuboids and LMP descriptors within the four proposed classification frameworks are presented in Fig. 5. The lowest error is achieved by the concatenated dictionary when used with the LMP descriptors and the resulting recognition accuracy is 98.9 percent (1 misclassification out of 90). The confusion matrices correspondiong to the two higher recognition results achieved in our experiments are presented in Fig. 6. In Table 2, the proposed approach is compared with a number of existing approaches, all of which use the leave-one-out scheme to evaluate their respective algorithms. Please note that some of the works use an older version of the Weizmann action data set which has nine classes of actions. We have used a later version of the data set that contains 10 classes. Our result achieves the highest accuracy among those which use the data set with 10 classes.

Synthetic Occlusion: We have also tested the robustness of our approach against occlusion. Since the original data set has no occlusion, we have selected a set of 10 action sequences

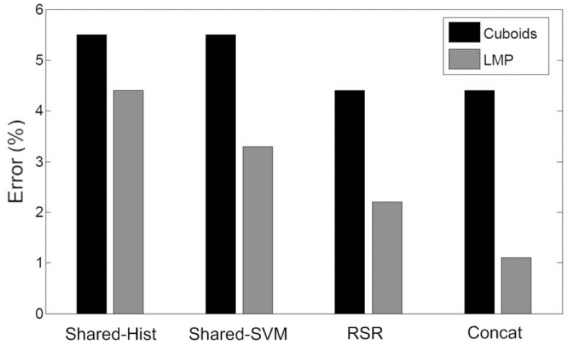
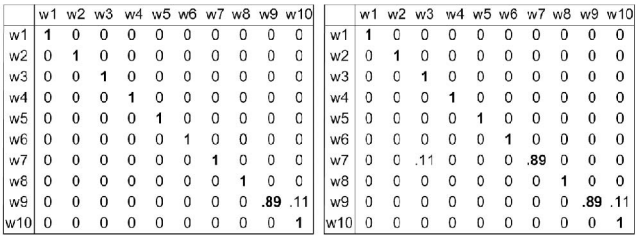


Fig. 5. Relative performances of the classification frameworks for Cuboids and LMP descriptors.



	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10
w1	1	0	0	0	0	0	0	0	0	0
w2	0	1	0	0	0	0	0	0	0	0
w3	0	0	1	0	0	0	0	0	0	0
w4	0	0	0	1	0	0	0	0	0	0
w5	0	0	0	0	1	0	0	0	0	0
w6	0	0	0	0	0	1	0	0	0	0
w7	0	0	0	0	0	0	1	0	0	0
w8	0	0	0	0	0	0	0	1	0	0
w9	0	0	0	0	0	0	0	0	.89	.11
w10	0	0	0	0	0	0	0	0	0	1

(a)

	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10
w1	1	0	0	0	0	0	0	0	0	0
w2	0	1	0	0	0	0	0	0	0	0
w3	0	0	1	0	0	0	0	0	0	0
w4	0	0	0	1	0	0	0	0	0	0
w5	0	0	0	0	1	0	0	0	0	0
w6	0	0	0	0	0	1	0	0	0	0
w7	0	0	.11	0	0	0	.89	0	0	0
w8	0	0	0	0	0	0	0	1	0	0
w9	0	0	0	0	0	0	0	0	.89	.11
w10	0	0	0	0	0	0	0	0	0	1

(b)

Fig. 6. (a) LMP + Concat. (mean accuracy 98.9 percent). (b) LMP + RSR (mean accuracy 97.8 percent).

TABLE 2
Comparison with State-of-the-Art on the Weizmann Action Data Set

Approach	No. of actions	Accuracy (%)
Yeffet & Wolf [29]	9	100
Wang & Mori [10]	9	100
Gorelick et al. [9]	10	97.8
Riemenschneider et al. [30]	10	96.7
Ali & Shah [31]	10	95.7
Junejo et al. [22]	9	95.3
Thurau & Hlavac [32]	10	94.4
Zhang et al. [33]	10	92.8
Simple reconstruction	10	92.2
Niebles et al. [21]	10	90.0
Scovanner et al. [17]	10	84.2
Our approach	10	**98.9**

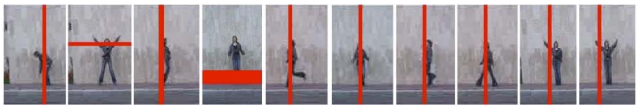


Fig. 7. Synthetic occlusion created by the authors of this paper.

TABLE 3
Results on the Weizmann Action Data Set: Performance under Synthetic Occlusion Using LMP Descriptors

Test sequence	Ground truth	Shared	RSR	Concat.
occluded by a pole	bend	bend	bend	bend
occluded by a bar	jack	jack	jack	jack
occluded by a pole	jump	jump	jump	jump
occluded feet	pjump	pjump	pjump	pjump
occluded by a pole	run	run	run	run
occluded by a pole	side	side	side	side
occluded by a pole	skip	skip	skip	skip
occluded by a pole	walk	walk	walk	walk
occluded by a pole	wave1	wave1	wave1	wave1
occluded by a pole	wave2	wave2	wave2	wave2

performed by one subject from the original data set and artificially created occlusion in all or some of the frames (refer to Fig. 7). Our approach achieves perfect accuracy under synthetic occlusion. The results are in Table 3.

Real Occlusion and Viewpoint Changes: There are 20 additional video sequences, known as the Weizmann Robustness data set, where the subjects walking in a nonuniform background create various difficult scenarios due to occlusion, clothing changes, unusual walking style, and viewpoint changes. Ten of the sequences exhibit viewpoint changes and the rest contain occlusion, etc. Sample frames can be found in Fig. 8. Our system is trained on the Weizmann action data set and is presented with the robustness sequences as queries.

Fig. 8. Sample frames from the Weizmann robustness data set showing occlusion, unusual scenarios, and viewpoint variations.

TABLE 4
Results on the Robustness Data Set: Performance under Real Occlusion and Other Difficult Scenarios Using LMP Descriptors (Trained on the Weizmann Action Data Set)

Test sequence	Gorelick et al. [9]	Shared	RSR	Concat.
walking with a dog	walk	walk	walk	walk
swinging a bag	walk	walk	walk	walk
walking in a skirt	walk	walk	walk	walk
occluded legs	walk	walk	walk	walk
occluded by a pole	walk	walk	walk	walk
normal walk	walk	walk	walk	walk
carrying briefcase	walk	walk	walk	walk
knees up	walk	run	walk	walk
limping walk	walk	walk	walk	walk
sleepwalking	walk	walk	walk	walk

TABLE 5
Results on the Robustness Data Set: Performance under Viewpoint Changes with the System Only Trained with Subjects Walking in 0° Using LMP Descriptors (Trained on the Weizmann Action Data Set)

Test sequence	Gorelick et al. [9]	Shared (svm)	RSR	Concat.
walking in 0°	walk	walk	walk	walk
walking in 9°	walk	walk	walk	walk
walking in 18°	walk	walk	walk	walk
walking in 27°	walk	walk	walk	walk
walking in 36°	walk	walk	walk	walk
walking in 45°	walk	walk	walk	walk
walking in 54°	walk	walk	walk	walk
walking in 63°	walk	skip	walk	walk
walking in 72°	walk	skip	walk	skip
walking in 81°	walk	side	skip	side

Tables 4 and 5 present the results under occlusion and viewpoint changes. Our results are compared with that reported in [9]. The RSR and concatenated dictionary demonstrates 100 percent accuracy against real occlusion and other difficult scenarios. Table 5 shows that, among others, the RSR algorithm exhibits maximum robustness against viewpoint changes. It correctly recognizes all except the sequence showing extreme viewpoint change, i.e., when the direction of walking in the test sequence is almost orthogonal to that in the training sequences. Recall that the system is trained with the sequences from the Weizmann action data set where the subjects are walking parallel to the camera, i.e., in 0 degree. The concatenated and shared dictionary-based methods are tolerant up to 63 and 54 degree changes in the viewpoint angle.

5.3 The Ballet Data Set

The ballet database is selected to test the ability of our approach to model complex motions. The data set contains 44 real video sequences of eight actions collected from an instructional ballet DVD [10], [34].[1] The eight actions

1. Both of these work address the problem of recognizing actions in still images which is a different problem altogether.

Fig. 9. (a) Sample frames from the Ballet data set: Left-to-right hand opening (b1), right-to-left hand opening (b2), standing hand opening (b3), leg swinging (b4), jumping (b5), turning (b6), hopping (b7), and standing still (b8).

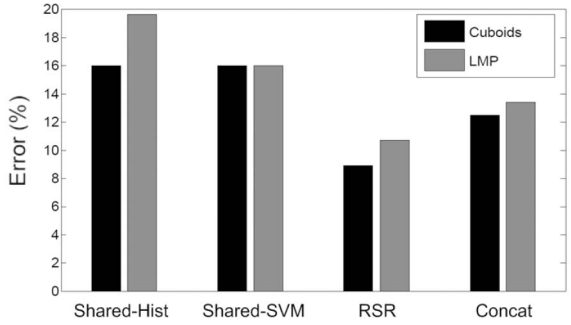


Fig. 10. Results on the Ballet data set: Relative performances of the classification frameworks for Cuboids and LMP descriptors.

	b1	b2	b3	b4	b5	b6	b7	b8
b1	**1**	0	0	0	0	0	0	0
b2	0	**.86**	0	0	0	.07	.07	0
b3	0	0	**1**	0	0	0	0	0
b4	0	0	0	**1**	0	0	0	0
b5	0	0	.07	0	**.64**	0	.29	0
b6	0	0	0	0	0	**1**	0	0
b7	0	0	.07	0	0	0	**.93**	0
b8	0	0	0	0	0	0	.14	**.86**

(a)

	b1	b2	b3	b4	b5	b6	b7	b8
b1	**1**	0	0	0	0	0	0	0
b2	0	**.86**	0	0	0	.07	.07	0
b3	0	0	**1**	0	0	0	0	0
b4	0	0	0	**1**	0	0	0	0
b5	0	0	.07	0	**.49**	.07	.37	0
b6	0	0	0	0	0	**1**	0	0
b7	0	0	.07	0	0	0	**.93**	0
b8	0	0	0	0	0	0	.14	**.86**

(b)

Fig. 11. Results on the Ballet data set: (a) Cuboids + RSR (91.1 percent). (b) LMP + RSR (89.3 percent).

TABLE 6
Comparison with State-of-the-Art on the Ballet Data Set

Approach	Accuracy (%) (frame based)	Accuracy (%) (video based)
Fathi & Mori [34]	51.0	-
Wang & Mori SLDA [10]	88.6	-
Wang & Mori SCTM [10]	91.3	-
Our approach	-	**91.1**

Note that [34] and [10] use a different experimental setup, so true comparison is not possible.

performed by three subjects are: left-to-right hand opening, right-to-left hand opening, standing hand opening, leg swinging, jumping, turning, hopping, and standing still. Fig. 9 presents the sample frames of each action. Ballet movements consist of complex motion patterns, the execution of which differs from performer to performer. The data set is highly challenging due to the significant intraclass variations in terms of speed, spatial, and temporal scale, clothing, and movement variations.

The results presented in Fig. 10 show that the RSR algorithm generates lower error rates for both Cuboids and LMP descriptors. Two confusion matrices are presented in Fig. 11. Maximum error is caused by the action "jumping" as it is confused with a very similar action "hopping." Table 6 compares our results with that of two previous papers which use this data set. Our method achieves comparable accuracy, but since [10], [34] perform image-based recognition and use a different experimental setup, the comparison is not a true

Fig. 12. Sample frames from the UCF Sports data set: diving (s1), golf swinging (s2), kicking (s3), lifting (s4), horse riding (s5), running (s6), skating (s7), swinging (s8), and walking (s9).

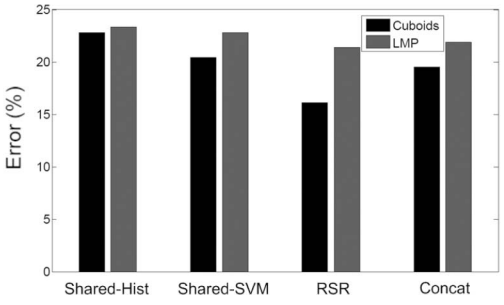


Fig. 13. Results on the UCF sports data set: Relative performances of the classification frameworks for Cuboids and LMP descriptors.

	s1	s2	s3	s4	s5	s6	s7	s8	s9
s1	**1**	0	0	0	0	0	0	0	0
s2	0	**.86**	0	0	0	0	0	0	.14
s3	0	0	**.48**	0	0	.24	.28	0	0
s4	0	0	0	**1**	0	0	0	0	0
s5	0	0	0	.14	**.43**	0	.14	0	.29
s6	0	0	0	0	.10	**.90**	0	0	0
s7	0	0	0	0	0	0	**.71**	0	.29
s8	0	0	0	0	0	0	0	**1**	0
s9	0	0	0	0	0	0	0	0	**1**

(a)

	s1	s2	s3	s4	s5	s6	s7	s8	s9
s1	**1**	0	0	0	0	0	0	0	0
s2	0	**1**	0	0	0	0	0	0	0
s3	.04	0	**.43**	0	0	.28	0	0	.25
s4	0	0	0	1	0	0	0	0	0
s5	0	.05	0	0	**.33**	.19	0	0	.43
s6	0	0	0	0	0	**1**	0	0	0
s7	0	0	0	0	0	.15	**.28**	0	.57
s8	0	0	0	0	0	0	0	**1**	0
s9	0	0	0	0	0	0	0	0	**1**

(b)

Fig. 14. Results on the UCF sports data set: (a) Cuboids + RSR (83.8 percent). (b) Cuboids + concat (80.9 percent).

one. Given the difficulty of the data set, the high recognition rate achieved by our approach is rather encouraging.

5.4 The UCF Sports Data Set

The UCF Sports data set [11] is considered to be one of the most challenging data sets in the field of action recognition. This data set contains close to 200 action sequences collected from various sports videos which are typically featured on broadcast television channels such as BBC and ESPN. The collection represents a natural pool of actions featured in a wide range of scenes and viewpoints. The data set also exhibits occlusion, cluttered background, variations in illumination, scale, and motion discontinuity. The nine actions are: diving, golf swinging, kicking, lifting, horse riding, running, skating, swinging, and walking (refer to Fig. 12). Some of these sequences also contain more than one subject.

The recognition results and confusion matrices are presented in Figs. 13 and 14. The highest accuracy achieved in our experiments is 83.8 percent (cuboids+RSR). Table 8 compares the proposed approach with a number of existing ones. Apparently, our recognition rate is lower than that reported in [15], [20], [35], and [36], but notice that, unlike [15], [20], we have not enlarged[2] the training set and used a much smaller dictionary. Also, [15], [20], and [36] use dense sampling with HoG3D descriptors as features. Such features are computationally more demanding compared to the features we have used. The result in [35] is obtained using

2. In [20] and [15], the data sets are enlarged by adding horizontally flipped version of each sequence.

TABLE 7
Comparison Using the Same Features on the UCF Sports Data Set

Approach	Feature	Classifier	Codewords/Atoms	Accuracy (%)
Wang et al. [20]	cuboids + HoG	non-linear SVM	4000	72.2
Our approach	cuboids + HoG	linear SVM	512	**79.6**
Our approach	cuboids + HoG	RSR	256 per dictionary	**83.8**

dense features, randomized trees, and Hough transform-based voting. This method is also computationally more intense compared to our approach. Both the features and the classification approach contribute to the recognition result. So, it is difficult to assess the contribution of our approach by comparing it with the methods that use different descriptors. In order to find out the real contribution of our sparse representation and classification approach, we concentrate on the results that are obtained using the same descriptors as ours. In Table 7, we compare our results with that of [20], which uses the same features as ours. Our approach shows significant improvement in accuracy (more than 10 percent). These results also serve as proof to that our sparse representation-based approach outperforms vector quantization-based methods in terms of accuracy and efficiency (note that our method also uses smaller dictionaries).

5.5 Facial Expression Data Set

The facial expression data set [8] involves two individuals, each expressing six different emotions under two lighting setups. The expressions are anger, disgust, fear, joy, sadness, and surprise, as shown in Fig. 15. Expressions such as sadness and joy are quite distinct but others are fairly similar, such as fear and surprise. Under each lighting setup, each individual shows each of the six expressions eight times. The subjects always start with a neutral expression, show an emotion, and return to neutral.

Fig. 16 presents the intraclass recognition results of the classification methods for Cuboids and LMP descriptors. The concatenated dictionary-based method produces the lowest errors for both types of descriptors. It is shown in [8] that for the facial expression data set, the concatenated gradient vector provides much better result compared to HoG. We have tested our approach with this descriptor so as to provide a true comparison with the original work in [8]. The results and comparison can be found in Fig. 17 and Table 9.

TABLE 8
Comparison with State-of-the-Art on the UCF Sports Data Set

Approach	Accuracy (%)
Rodriguez et al. [11]	69.2
Yeffet & Wolf [29]	79.2
Zhu et al. [15]	84.3
Wang et al. [20]	85.6
Yao et al. [35]	86.6
Kovashka & Grauman [36]	87.2
Our approach	**83.8**

Fig. 15. Sample frames from the facial expression data set: anger (f1), disgust (f2), fear (f3), joy (f4), sadness (f5), and surprise (f6).

5.6 Remarks

A few interesting observations can be made from the experimental results:

- Our proposed sparse modeling approach significantly outperforms the traditional vector-quantization-based BoW modeling. In Table 7, the proposed approach shows more than 10 percent improvement over its vector quantization-based counterpart. It is also confirmed in Table 9.
- The class-specific dictionaries (or their concatenation) produce better recognition results compared to

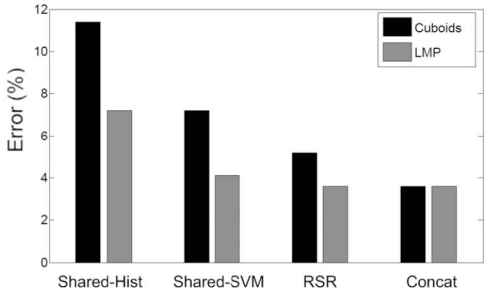


Fig. 16. Results on the facial expression data set: Relative performances of the classification frameworks for Cuboids and LMP descriptors.

	f1	f2	f3	f4	f5	f6
f1	**1**	0	0	0	0	0
f2	0	**1**	0	0	0	0
f3	0	.25	**.75**	0	0	0
f4	0	0	0	**1**	0	0
f5	0	0	0	0	**1**	0
f6	0	0	.25	0	0	**.75**

(a)

	f1	f2	f3	f4	f5	f6
f1	**1**	0	0	0	0	0
f2	.25	**.63**	.12	0	0	0
f3	0	0	**.5**	0	0	.5
f4	0	0	0	**1**	0	0
f5	.25	0	0	0	**.75**	0
f6	0	.25	.25	0	0	**.5**

(b)

Fig. 17. Results on the facial expression data set: (a) Different subject, same illumination (91.7 percent). (b) Different subject, different illumination (72.9 percent).

TABLE 9
Comparison on the Facial Expression Data Set

Condition	Accuracy (%) Dollar et al. [8]	Accuracy (%) (Our approach)
same subject & lighting	97.9	**100**
same subject, different lighting	89.6	**93.7**
different subject, same lighting	75.0	**91.7**
different subject & lighting	69.8	**72.9**

The sparse modeling approach is compared with the traditional BoW approach. For true comparison, we have used the exact detector and descriptor specifications used in [8].

the shared dictionaries. We advocate the use of class-specific dictionaries because along with superior results they also offer ways to save computation (mentioned in Section 4.5). Both RSR and concatenated methods work well, but RSR appears to be more stable and consistent. RSR, being robust to outliers, can better deal with complex data sets like Ballet or UCF sports.

- RP practically overrules the use of traditional dimensionality reduction methods like PCA within this framework. It is the fastest possible dimensionality reduction process. It also keeps the dictionary dimension more manageable. There is no direct relationship between the feature dimension and codebook size in vector-quantization-based BoW modeling. While working with sparse representation, the overcompleteness factor is at least 2, i.e., the number of atoms in a dictionary is at least twice the size of the features. For a given feature dimension, increasing the overcompleteness factor (i.e., increasing the number of atoms) does not necessarily increase the accuracy, but it does raise the cost of computation. From our experiments, we found that increasing the overcompleteness factor beyond 4 does not improve the results much and in fact, starts to fall for overcompleteness factors greater than or equal to 6.
- It is interesting to notice that simple features like LMP can outperform sophisticated features like Cuboids in some cases. This can be attributed to the fact that LMP, being based on 2D keypoint detectors, generates some static features (in the regions having distinctive spatial structure but not undergoing much temporal change). The static features are known to contain useful information for recognition.

6 Conclusion

This work studies the usefulness of sparse representations obtained using learned overcomplete dictionaries in the context of video-based action modeling and recognition. The ideas proposed in this paper are fairly general and are applicable to other recognition problems, such as object recognition. Experimental results demonstrate that the proposed approach is computationally efficient, highly accurate, and is robust against partial occlusion, spatio-temporal scale variations, and to some extent to viewpoint changes. This robustness is achieved by exploiting the discriminative nature of the sparse representations combined with spatio-temporal motion descriptors. The fact that the descriptors are extracted over multiple temporal and spatial resolutions make them insensitive to scale changes. The descriptors being computed locally make them robust against occlusion or other distortions.

We have used OMP—the simplest pursuit algorithm to solve all the sparse approximation problems. More sophisticated solvers, e.g., BP can achieve better results but at the cost of higher computation time. Likewise, features such as dense sampling [19], HoG3D, STIP [16], etc., can also improve the recognition accuracy but are more expensive computationally.

Our system at present cannot deal with multiple actions presented in one video sequence. This is because we disregard the spatial and temporal orientation of the extracted features. Incorporating such information will help detecting and recognizing multiple actions. Other future works include learning hierarchical dictionaries, discriminative dictionaries, and building dictionaries using different descriptors or a combination of them. Also, techniques are required to optimize parameters like sparsity, dictionary size, etc.

Acknowledgments

The authors are grateful to the anonymous reviewers whose insightful comments and suggestions have improved the quality of this paper. They would like to thank Dr. Piotr Dollár, University of California, San Diego, who readily provided the code for the Cuboids descriptor. Also, thanks to Professor Luis Torres, Technical University of Catalonia, Barcelona, Spain, for his helpful comments that improved an earlier version of this paper.

References

[1] B. Wohlberg, "Noise Sensitivity of Sparse Signal Representations: Reconstruction Error Bounds for the Inverse Problem," *IEEE Trans. Signal Processing,* vol. 51, no. 12, pp. 3053-3060, Dec. 2003.

[2] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, and Y. Ma, "Robust Face Recognition via Sparse Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 2, pp. 210-227, Feb. 2008.

[3] M.S. Lewicki and T.J. Sejnowski, "Learning Overcomplete Representations," *Neural Computation,* vol. 12, no. 2, pp. 337-365, 2000.

[4] K. Engan, S.O. Aase, and J.H. Husoy, "Method of Optimal Directions for Frame Design," *Proc. IEEE Int'l Conf. Audio, Speech and Signal Processing,* 1999.

[5] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Trans. Signal Processing,* vol. 54, no. 11, pp. 4311-4322, Nov. 2006.

[6] J. Mairal, M. Elad, and G. Sapiro, "Sparse Representation for Color Image Restoration," *IEEE Trans. Image Processing,* vol. 17, no. 1, pp. 53-69, Jan. 2008.

[7] M. Elad and M. Aharon, "Image Denoising via Sparse and Redundant Representations over Learned Dictionaries," *IEEE Trans. Image Processing,* vol. 15, no. 12, pp. 3736-3745, Dec. 2006.

[8] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior Recognition via Sparse Spatio-Temporal Features," *Proc. Second IEEE Joint Int'l Workshop Visual Surveillance and Performance Evaluation of Tracking and Surveillance,* pp. 65-72, Oct. 2005.

[9] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as Space-Time Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 12, pp. 2247-2253, Dec. 2007.

[10] Y. Wang and G. Mori, "Human Action Recognition by Semilatent Topic Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 10, pp. 1762-1774, Oct. 2009.

[11] M. Rodriguez, J. Ahmed, and M. Shah, "Action Match a Spatio-Temporal Maximum Average Correlation Height Filter for Action Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, June 2008.

[12] G. Peyré, "Sparse Modeling of Textures," *J. Math. Imaging and Vision,* vol. 34, no. 1, pp. 17-31, 2009.

[13] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative Learned Dictionaries for Local Image Analysis," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[14] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1794-1801, 2009.

[15] Y. Zhu, X. Zhao, Y. Fu, and Y. Liu, "Sparse Coding on Local Spatial-Temporal Volumes for Human Action Recognition," *Proc. 10th Asian Conf. Computer Vision,* vol. 6493, pp. 660-671, 2010.
[16] I. Laptev, "On Space-Time Interest Points," *Int'l J. Computer Vision,* vol. 64, pp. 107-123, 2005.
[17] P. Scovanner, S. Ali, and M. Shah, "A 3-Dimensional Sift Descriptor and Its Application to Action Recognition," *Proc. 15th Int'l Conf. Multimedia,* pp. 357-360, 2007.
[18] A.A. Efros, A.C. Berg, G. Mori, and J. Malik, "Recognizing Action at a Distance," *Proc. Ninth IEEE Int'l Conf. Computer Vision,* 2003.
[19] F. Jurie and B. Triggs, "Creating Efficient Codebooks for Visual Recognition," *Proc. 10th IEEE Int'l Conf. Computer Vision,* 2005.
[20] H. Wang, M.M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of Local Spatio-Temporal Features for Action Recognition," *Proc. British Machine Vision Conf.,* Sept. 2009.
[21] J. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words," *Int'l J. Computer Vision,* vol. 79, pp. 299-318, 2008.
[22] I.N. Junejo, E. Dexter, I. Laptev, and P. Perez, "View-Independent Action Recognition from Temporal Self-Similarities," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 33, no. 1, pp. 172-185, Jan. 2011.
[23] A.F. Bobick and J.W. Davis, "The Recognition of Human Movement Using Temporal Templates," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 3, pp. 257-267, Mar. 2001.
[24] D.G. Lowe, "Distinctive Image Features from Scale-invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, pp. 91-110, 2004.
[25] R. Baraniuk and M. Wakin, "Random Projections of Smooth Manifolds," *Foundations of Computational Math.,* vol. 9, pp. 51-77, 2009.
[26] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM,* vol. 24, pp. 381-395, June 1981.
[27] Y. Eldar and H. Bolcskei, "Block-Sparsity: Coherence and Efficient Recovery," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,* pp. 2885-2888, 2009.
[28] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of Interest Point Detectors," *Int'l J. Computer Vision,* vol. 37, pp. 151-172, 2000.
[29] L. Yeffet and L. Wolf, "Local Trinary Patterns for Human Action Recognition," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 492-497, Oct. 2009.
[30] M.D. Hayko Riemenschneider and H. Bischof, "Bag of Optical Flow Volumes for Image Sequence Recognition," *Proc. British Machine Vision Conf.,* 2009.
[31] S. Ali and M. Shah, "Human Action Recognition in Videos Using Kinematic Features and Multiple Instance Learning," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 2, pp. 288-303, Feb. 2010.
[32] C. Thurau and V. Hlavac, "Pose Primitive Based Human Action Recognition in Videos or Still Images," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, 2008.
[33] Z. Zhang, Y. Hu, S. Chan, and L.-T. Chia, "Motion Context: A New Representation for Human Action Recognition," *Proc. European Conf. Computer Vision,* vol. 5305, pp. 817-829, 2008.
[34] A. Fathi and G. Mori, "Action Recognition by Learning Mid-Level Motion Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, 2008.
[35] A. Yao, J. Gall, and L. Van Gool, "A Hough Transform-based Voting Framework for Action Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 2061-2068, June 2010.
[36] A. Kovashka and K. Grauman, "Learning a Hierarchy of Discriminative Space-Time Neighborhood Features for Human Action Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 2046-2053, June 2010.

Tanaya Guha is currently working toward the PhD degree in electrical and computer engineering at the University of British Columbia, Vancouver, British Columbia, Canada. Her research interests include image and video processing, pattern recognition, and machine learning. She is currently conducting research on overcomplete dictionary learning and its applications in classification. She is a student member of the IEEE.

Rabab Kreidieh Ward is a professor in the Electrical and Computer Engineering Department at the University of British Columbia, Vancouver, British Columbia, Canada. She is presently with the Office of the V.P. Research and International as the natural sciences and engineering research coordinator. Her research interests include signal, image, and video processing. She has made contributions in the areas of signal detection, image encoding, compression, recognition, restoration, and enhancement, and their applications to infant cry signals, cable TV, HDTV, medical images, and astronomical images. She has published approximately 400 refereed journal and conference papers and book chapters. She holds six patents related to cable television picture monitoring, measurement, and noise reduction. Applications of her work have been transferred to US and Canadian industries. She has served as the VP of the IEEE Signal Processing Society, the general chair of the IEEE ICIP 2000, IEEE ISSPIT 2006, and the vice chair of the IEEE ISCAS 2004. She is a fellow of the Royal Society of Canada, the IEEE, the Canadian Academy of Engineers, and the Engineering Institute of Canada. She is a recipient of the UBC Killam Research Prize, the YWCA Woman of Distinction Award, the R.A. McLachlan Memorial Award (the top award) of the Association of Professional Engineers and Geoscientists of British Columbia, the "Society Award" of the IEEE Signal Processing Society, and the 2012 Career Achievement Award of the Confederation of University Faculty Associations of British Columbia.