Action Recognition Using Super Sparse Coding Vector with Spatio-temporal Awareness

Xiaodong Yang and YingLi Tian

Department of Electrical Engineering
City College, City University of New York, USA

Abstract. This paper presents a novel framework for human action recognition based on sparse coding. We introduce an effective coding scheme to aggregate low-level descriptors into the super descriptor vector (SDV). In order to incorporate the spatio-temporal information, we propose a novel approach of super location vector (SLV) to model the space-time locations of local interest points in a much more compact way compared to the spatio-temporal pyramid representations. SDV and SLV are in the end combined as the super sparse coding vector (SSCV) which jointly models the motion, appearance, and location cues. This representation is computationally efficient and yields superior performance while using linear classifiers. In the extensive experiments, our approach significantly outperforms the state-of-the-art results on the two public benchmark datasets, i.e., HMDB51 and YouTube.

1 Introduction

Action recognition has been widely applied to a number of real-world applications, e.g., surveillance event detection, human-computer interaction, content-based video search, etc. It is of great challenge to recognize actions in unconstrained videos due to the large intra-class variations caused by factors such as viewpoint, occlusion, motion style, performance duration, and cluttered background. Most of recent action recognition approaches rely on the bag-of-visual-words (BOV) representation which consists in computing and aggregating statistics from local space-time features [15] [26] [27]. In this framework, a video representation can be obtained by extracting low-level features, coding them over a visual dictionary, and pooling the codes in some well-chosen support regions. A significant progress has been made in the development of local space-time features [26] [27]. After low-level feature extraction, the approaches similar to those used for object recognition are generally employed.

In the basic BOV framework, a visual dictionary is learned by K-means and used to quantize low-level features through hard-assignment [15]. A number of coding variants have been proposed and reported to achieve the state-of-the-art results in image and video recognition, e.g., local soft assignment [19], sparse coding [30], and locality-constrained linear coding [28]. These approaches reduce information loss by relaxing the restrictive cardinality constraint in coding

D. Fleet et al. (Eds.): ECCV 2014, Part II, LNCS 8690, pp. 727–741, 2014.

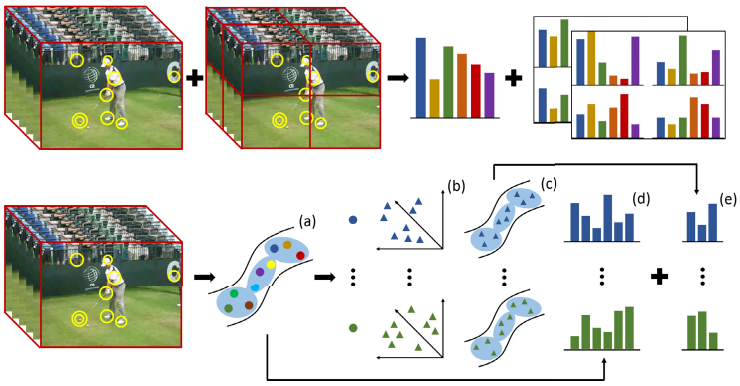

Fig. 1. Frameworks of STP (up) and SSCV (bottom). STP represents a video by concatenating BOVs from the entire sequence and spatio-temporal cells. SSCV jointly models the motion, appearance, and location information. (a) A visual dictionary of local descriptors is learned by sparse coding. (b) 3D space-time locations are associated to each visual word in (a) according to the assignments of descriptors. (c) A visual dictionary of locations is learned by sparse coding for each set in (b). SSCV is obtained by the combination of (d) SDV and (e) SLV.

descriptors. Accordingly, the average pooling can be replaced by the max pooling [30]. Recently, several coding schemes have emerged to encode descriptors with respect to the visual words that they are assigned to, e.g., Fisher vector [23], super vector coding [31], and vector of locally aggregated descriptors [10]. These methods usually retain high order statistics and have noticeably better performances [25].

The basic BOV aggregates the assignments over an entire video to generate the final representation. It obviously incurs a loss of information by discarding all the spatio-temporal locations of local space-time features. An extension to the completely orderless BOV for action recognition is the spatio-temporal pyramid (STP) [15] [26], inspired by the spatial pyramid matching (SPM) [16] for image classification. In this approach, a video sequence is repeatedly and evenly subdivided into a set of spatial and temporal cells where descriptor-level statistics are pooled. It can be used to roughly capture the spatial layout and temporal order of an action sequence. However, the concatenation of BOV histograms over many subvolumes of a video dramatically increases feature dimensions, which largely increase the learning and storage costs.

In this paper, we propose a novel action recognition framework on low-level feature coding and spatio-temporal information modeling, as illustrated in Fig. 1. We first employ a sparse coding approach [20] to compute the visual dictionary and coefficients of local descriptors. Each descriptor is coded by recording the difference of the local descriptor to all visual words. The coefficient-weighted difference vectors are then aggregated for each visual word through the whole video. These vectors of all visual words are in the end concatenated as the representation of super descriptor vector (SDV), which is used to characterize the

motion and appearance cues. We further model the spatio-temporal information by computing the super location vector (SLV) of the space-time coordinates of local descriptors assigned to each visual word. We combine SDV and SLV as the super sparse coding vector (SSCV) which jointly models the motion, appearance, and spaio-temporal information.

The main contributions of this paper are summarized as follows. First, we provide an effective coding scheme to aggregate low-level features into a discriminative representation, which relies on a smaller visual dictionary. Second, we propose a novel approach to incorporate the spaio-temporal information in a much more compact representation, which correlates and models the motion, appearance, location cues in a unified way. Third, we perform a systematic evaluation of the state-of-the-art coding and pooling methods in the context of action recognition.

The remainder of this paper is organized as follows. Section 2 introduces the related work of feature aggregation and spatio-temporal information modeling. Section 3 describes the detailed procedures to compute SDV, SLV, and SSCV. A variety of experimental results and discussions are presented in Section 4. Finally, Section 5 summarizes the remarks of this paper.

2 Notations and Related Work

In this section, we introduce the notations used throughout this paper and summarize the related work on aggregating local descriptors and modeling spatial (temporal) information. We represent a video sequence $\mathcal{V}$ by a set of low-level descriptors $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ in $\mathbb{R}^{m \times n}$ and associated locations $\mathcal{L} = \{\boldsymbol{l}_1, \ldots, \boldsymbol{l}_n\}$ in $\mathbb{R}^{3 \times n}$. C indicates the space-time cells defined in a spatio-temporal pyramid with C_j denoting the jth cell. $\mathbf{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_K\}$ is a visual dictionary with K visual words $\mathbf{d}_k \in \mathbb{R}^m$.

2.1 Feature Aggregation

Let $\mathcal{F}$ and $\mathcal{G}$ denote the coding and pooling operators, respectively. The final representation of $\mathcal{V}$ is the vector $\boldsymbol{z}$ obtained by sequentially coding, pooling, and concatenating over all space-time cells:

$$\boldsymbol{\alpha}_i = \mathcal{F}(\boldsymbol{x}_i), \quad i = 1, \ldots, n, \tag{1}$$

$$\boldsymbol{h}_j = \mathcal{G}\left(\{\boldsymbol{\alpha}_i\}_{i \in C_j}\right), \quad j = 1, \ldots, |C|, \tag{2}$$

$$\boldsymbol{z}^T = \left[\boldsymbol{h}_1^T \ldots \boldsymbol{h}_{|C|}^T\right]. \tag{3}$$

In the basic BOV framework, hard assignment $\mathcal{F}$ minimizes the distance of $\boldsymbol{x}_i$ to $\mathbf{D}$ that is usually learned by K-means. $\mathcal{G}$ performs the averaging over each pooling cell C_j:

$$\boldsymbol{\alpha}_i \in \{0,1\}^K, \boldsymbol{\alpha}_{i,j} = 1 \text{ iff } j = \arg\min_k \|\boldsymbol{x}_i - \mathbf{d}_k\|_2^2, \tag{4}$$

$$\boldsymbol{h}_j = \frac{1}{|C_j|} \sum_{i \in C_j} \boldsymbol{\alpha}_i. \tag{5}$$

In order to enhance the probability density estimation, soft assignment was introduced in [5]. It codes a descriptor $\boldsymbol{x}_i$ by multiple visual words in $\mathbf{D}$ using a kernel function (e.g., the Gaussian function) of the distance between $\boldsymbol{x}_i$ and $\mathbf{d}_k$. Liu et al. proposed local soft assignment in [19] to further improve the membership estimation to visual words. By taking account of the underlying manifold structure of local descriptors, $\mathcal{F}$ in local soft assignment only employs the $\mathcal{K}$ nearest visual words $N_{\mathcal{K}}(\boldsymbol{x}_i)$ to code a descriptor $\boldsymbol{x}_i$ and sets its distances of the remaining visual words to infinity:

$$\boldsymbol{\alpha}_{i,k} = \frac{\exp\left(-\beta \hat{d}\left(\boldsymbol{x}_i, \mathbf{d}_k\right)\right)}{\sum_{j=1}^{K} \exp\left(-\beta \hat{d}\left(\boldsymbol{x}_i, \mathbf{d}_j\right)\right)}, \tag{6}$$

$$\hat{d}\left(\boldsymbol{x}_i, \mathbf{d}_k\right) = \begin{cases} \|\boldsymbol{x}_i - \mathbf{d}_k\|^2 & \text{if } \mathbf{d}_k \in N_{\mathcal{K}}\left(\boldsymbol{x}_i\right), \\ \infty & \text{otherwise,} \end{cases} \tag{7}$$

where β is a smoothing factor to control the softness of assignment. As for $\mathcal{G}$ in local soft assignment, it was observed that max pooling in the following equation outperformed average pooling:

$$\boldsymbol{h}_{j,k} = \max_{i \in C_j} \boldsymbol{\alpha}_{i,k}, \quad \text{for } k = 1, \ldots, K. \tag{8}$$

Parsimony has been widely employed as a guiding principle to compute sparse representation with respect to an overcomplete visual dictionary. Sparse coding [20] approximates $\boldsymbol{x}_i$ by using a linear combination of a limited number of visual words. It is well known that the ℓ_1 penalty yields a sparse solution. So the sparse coding problem can be solved by:

$$\min_{\mathbf{D}, \boldsymbol{\alpha}} \frac{1}{n} \sum_{i=1}^{n} \left(\frac{1}{2} \|\boldsymbol{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_1 \right), \tag{9}$$

$$\text{subject to } \mathbf{d}_k^T \mathbf{d}_k \leq 1, \forall k = 1, \ldots, K,$$

where λ is the sparsity-inducing regularizer to control the number of non-zero coefficients in $\boldsymbol{\alpha}_i$. It is customary to combine sparse coding with max pooling as shown in Eq. (8).

Fisher vector [23] extends the BOV representation by recording the deviation of $\boldsymbol{x}_i$ with respect to the parameters of a generative model, e.g., the Gaussian mixture model (GMM) characterized by $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k, k = 1, \ldots, K\}$. π_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\sigma}_k$ are the prior mode probability, mean vector, and covariance matrix (diagonal), respectively. Let γ_i^k be the soft assignment of $\boldsymbol{x}_i$ to the kth Gaussian

component. We obtain the Fisher vector of $\mathcal{X}$ by concatenating the gradient vectors from K Gaussian components:

$$\boldsymbol{\rho}_k = \frac{1}{n\sqrt{\pi_k}} \sum_{i=1}^{n} \gamma_i^k \left(\frac{\boldsymbol{x}_i - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k} \right), \quad \boldsymbol{\tau}_k = \frac{1}{n\sqrt{2\pi_k}} \sum_{i=1}^{n} \gamma_i^k \left[\frac{(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^2} - 1 \right], \quad (10)$$

where $\boldsymbol{\rho}_k$ and $\boldsymbol{\tau}_k$ are m-dimensional gradients with respect to $\boldsymbol{\mu}_k$ and $\boldsymbol{\sigma}_k$ of the kth Gaussian component. The relative displacements of descriptors to the mean and variance in Eq. (10) retain more information lost in the traditional coding process. The superiority of Fisher vector was recently identified in both image classification [25] and action recognition [29].

2.2 Spatial and Temporal Information

The orderless representation of a video completely ignores the spatial and temporal information, which could convey discriminative cues for action recognition. We outline the relevant representative work that attempts to account for the spatial and temporal locations of low-level features.

The dominant approach to incorporate spatial and temporal information is the spatio-temporal pyramid (STP), as illustrated in Fig. 1. Inspired by the spatial pyramid matching (SPM) [16], Laptev et al. [15] proposed to partition a video to a set of space-time cells in a coarse-to-fine manner. Each cell is represented independently and the cell-level histograms $\boldsymbol{h}_j$ are finally concatenated into the video-level histogram $\boldsymbol{z}$ as in Eq. (2-3). This representation has been proven to be effective when the action categories exhibit characteristic spatial layout and temporal order.

In image classification, the feature augmentation based methods were proposed in [21] [24] to append a weighted location $\boldsymbol{l}_i$ to the corresponding descriptor $\boldsymbol{x}_i$. As opposed to SPM, this approach does not increase the feature dimensionality thus makes the learning more efficient. Krapac et al. [12] introduced the spatial Fisher vector to encode the spatial layout of local image features. The location model can be learned by computing per visual word the mean and variance of spatial coordinates of the assigned local image patches. While these representations are more compact, the evaluation results only showed marginal improvement over SPM in terms of classification accuracy.

3 Super Sparse Coding Vector

We describe the detailed procedures of computing SSCV in this section. We propose a novel feature coding scheme based on sparse coding to aggregate descriptors and locations into discriminative representations. The space-time locations are included as part of the coding step, instead of only coding motion and appearance cues and leaving the spatio-temporal coherence to be represented in the pooling stage. This enables SSCV to jointly characterize the motion, appearance, and location information.

3.1 Modeling Space-Time Features

We represent each local feature as the descriptor-location tuple $\boldsymbol{f}_i = (\boldsymbol{x}_i, \boldsymbol{l}_i)$. By employing a generative model (e.g., GMM) over descriptors and locations, we model $\boldsymbol{f}_i$ as:

$$p(\boldsymbol{f}_i) = \sum_{k=1}^{K} p(w=k)p(\boldsymbol{x}_i|w=k)p(\boldsymbol{l}_i|w=k), \tag{11}$$

where $p(w=k)$ denotes the prior mode probability of the kth Gaussian component in the descriptor mixture model, and w is the assignment index. We assume the prior mode probabilities are equal, i.e., $p(w=k) = 1/K, \forall k$. The kth Gaussian of descriptors is defined by:

$$p(\boldsymbol{x}_i|w=k) = \mathcal{N}\left(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k\right), \tag{12}$$

where $\boldsymbol{\mu}_k$ and $\boldsymbol{\sigma}_k$ are the mean and covariance (diagonal) of the kth Gaussian. As illustrated in Fig. 1, we jointly model the spatio-temporal information by associating the locations of descriptors to the corresponding visual descriptor word, i.e., the Gaussian of descriptors in this context. We define the spatio-temporal model by using a GMM distribution over the locations associated with the kth visual word:

$$p(\boldsymbol{l}_i|w=k) = \sum_{g=1}^{G} \pi_{k_g} \mathcal{N}\left(\boldsymbol{l}_i; \boldsymbol{\mu}_{k_g}, \boldsymbol{\sigma}_{k_g}\right), \tag{13}$$

where π_{k_g}, $\boldsymbol{\mu}_{k_g}$, and $\boldsymbol{\sigma}_{k_g}$ are the prior mode probability, mean, and covariance (diagonal) of the gth Gaussian of locations in the kth visual descriptor word. We again assume the prior mode probabilities are equal, i.e., $\pi_{k_g} = 1/G, \forall g$.

3.2 Computing Super Descriptor Vector (SDV)

We utilize sparse coding to learn a visual dictionary and code descriptors. We aggregate the coefficient-weighted differences between local descriptors and visual words into a vector, rather than directly pooling the coefficients.

The generation process of $\boldsymbol{x}_i$ is modeled by the probability density function in Eq. (12). The gradient of the log-likelihood of this function with respect to its parameters describes the contribution of the parameters to the generation process [8]. Here we focus on the gradient with respect to the mean:

$$\frac{\partial \ln p(\boldsymbol{x}_i|w=k)}{\partial \boldsymbol{\mu}_k} = \rho_i^k \boldsymbol{\sigma}_k^{-1}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_k\right), \tag{14}$$

where ρ_i^k denotes the posterior $p(w=k|\boldsymbol{x}_i)$. If making the three approximations:

1. the posterior is estimated by the sparse coding coefficient, i.e., $\rho_i^k = \alpha_i^k$,
2. the mean is represented by the visual word in sparse coding, i.e., $\boldsymbol{\mu}_k = \mathbf{d}_k$,
3. the covariance is isotropic, i.e., $\boldsymbol{\sigma}_k = \epsilon\mathbb{I}$ with $\epsilon > 0$,

we can simplify Eq. (14) to $\alpha_i^k (\boldsymbol{x}_i - \mathbf{d}_k)$, where α_i^k is the coefficient of the ith descriptor $\boldsymbol{x}_i$ to the kth visual word $\mathbf{d}_k$ in Eq. (9).

We choose sparse coding in the approximation because it is much cheaper to compute the means (dictionary) compared to the Expectation Maximization (EM) algorithm in training GMM. Especially, it was recently shown in [3] that a reasonably good dictionary can be created by some simple methods, e.g., random sampling in a training set. Moreover, our empirical evaluations show the approximations based on sparse coding improves the recognition accuracy. We then apply average pooling to aggregate the coefficient-weighted difference vectors for each visual word:

$$\boldsymbol{u}_k = \frac{1}{n} \sum_{i=1}^{n} \alpha_i^k (\boldsymbol{x}_i - \mathbf{d}_k). \tag{15}$$

The final vector representation $\boldsymbol{U}$ of SDV is the concatenation of $\boldsymbol{u}_k$ from K visual words and is therefore with the dimensionality of mK:

$$\boldsymbol{U} = \left[\boldsymbol{u}_1^T \ldots \boldsymbol{u}_K^T\right]^T. \tag{16}$$

SDV has several remarkable properties: (1) the relative displacements of descriptors to visual words retain more information lost in the traditional coding process; (2) we can compute SDV upon a much smaller dictionary which reduces the computational cost; (3) it performs quite well with simple linear classifiers which are efficient in terms of both training and testing.

3.3 Computing Super Location Vector (SLV)

The descriptors quantized to the same visual word exhibit characteristic spatio-temporal layout. In order to capture this correlation between motion, appearance, and location, we associate space-time locations to the visual descriptor words that corresponding descriptors are assigned to. We also employ sparse coding to learn a visual location dictionary to code the location set associated with each visual descriptor word, as illustrated in Fig. 1(c). The coefficient-weighted differences between locations and visual location words are aggregated as the spatio-temporal representation.

In order to describe the contribution of the parameters to the generation process of $\boldsymbol{l}_i$, we take the gradient of the log-likelihood of Eq. (13) with respect to the mean:

$$\frac{\partial \ln p(\boldsymbol{l}_i | w = k)}{\partial \boldsymbol{\mu}_{k_g}} = \rho_i^{k_g} \boldsymbol{\sigma}_{k_g}^{-1} \left(\boldsymbol{l}_i - \boldsymbol{\mu}_{k_g}\right), \tag{17}$$

where $\rho_i^{k_g}$ denotes posterior $p(t = g | \boldsymbol{l}_i, w = k)$ and t is the assignment index. We can interpret $\rho_i^{k_g}$ as a spatio-temporal soft assignment of a descriptor location $\boldsymbol{l}_i$ associated with the kth visual descriptor word to the gth Gaussian component in the location mixture model.

Algorithm 1. Computation of SSCV

Input: a video sequence $\mathcal{V}$
a visual descriptor dictionary $\mathbf{D}^x = \{\mathbf{d}_k\}$
a visual location dictionary $\mathbf{D}^l = \{\mathbf{d}_{k_g}\}$
Output: SSCV $\boldsymbol{Z}$

compute spatio-temporal features $\mathcal{X} = \{\boldsymbol{x}_i\}$ and $\mathcal{L} = \{\boldsymbol{l}_i\}$ from $\mathcal{V}$
compute coefficients $\{\alpha_i^k\}$ of $\mathcal{X}$ on $\{\mathbf{d}_k\}_{k=1}^K$ by sparse coding
for visual descriptor word $k = 1$ **to** K **do**
  $\boldsymbol{u}_k :=$ average pooling $\alpha_i^k (\boldsymbol{x}_i - \mathbf{d}_k)$, $\boldsymbol{x}_i \in \mathcal{X}$
  associate locations to the kth visual descriptor word: $\mathcal{L}_k = \{\boldsymbol{l}_i | \alpha_i^k > 0\}$
  compute coefficients $\left\{\alpha_i^{k_g}\right\}$ of $\mathcal{L}_k$ on $\{\mathbf{d}_{k_g}\}_{g=1}^G$ by sparse coding
  for visual location word $g = 1$ **to** G **do**
    $\boldsymbol{v}_{k_g} :=$ average pooling $\alpha_i^{k_g} (\boldsymbol{l}_i - \mathbf{d}_{k_g})$, $\boldsymbol{l}_i \in \mathcal{L}_k$
  end
  $\boldsymbol{Z}_k := \left[\boldsymbol{u}_k^T, \boldsymbol{v}_{k_1}^T \dots \boldsymbol{v}_{k_G}^T\right]^T$
end
$\boldsymbol{Z} := \left[\boldsymbol{Z}_1^T \dots \boldsymbol{Z}_K^T\right]^T$
signed square rooting and ℓ_2 normalization

If we enforce the approximations (1-3) in Section 3.2, Eq. (17) can be simplified to $\alpha_i^{k_g} \left(\boldsymbol{l}_i - \mathbf{d}_{k_g}\right)$, where $\alpha_i^{k_g}$ is the sparse coding coefficient of the ith location $\boldsymbol{l}_i$ to the gth visual location word $\mathbf{d}_{k_g}$ associated with the kth visual descriptor word $\mathbf{d}_k$. As illustrated in Fig. 1(b), let $\mathcal{L}_k$ denote the set of locations that are associated to the kth visual descriptor word according to the positive assignments of their descriptors, i.e., $\mathcal{L}_k = \{\boldsymbol{l}_i | \alpha_i^k > 0\}$. We then employ the average pooling to aggregate the coefficient-weighted difference vectors for each visual location word:

$$\boldsymbol{v}_{k_g} = \frac{1}{|\mathcal{L}_k|} \sum_{\boldsymbol{l}_i \in \mathcal{L}_k} \alpha_i^{k_g} \left(\boldsymbol{l}_i - \mathbf{d}_{k_g}\right). \tag{18}$$

The concatenation of $\boldsymbol{v}_{k_g}$ from G visual location words associated with K visual descriptor words forms the final representation $\boldsymbol{V}$ of SLV with a dimensionality of $3GK$:

$$\boldsymbol{V} = \left[\boldsymbol{v}_{1_1}^T \dots \boldsymbol{v}_{1_G}^T \dots \boldsymbol{v}_{K_1}^T \dots \boldsymbol{v}_{K_G}^T\right]^T. \tag{19}$$

SLV shares the same remarkable properties as SDV. Moreover, SLV can be computed on much smaller visual descriptor dictionary (e.g., $K = 100$) and visual location dictionary (e.g., $G = 5$). If we combine SDV and SLV, the resulting vector is of $(m + 3G)K$ dimensions, where the descriptor dimensionality m (e.g., 162 in STIP [14]) is normally much larger than $3G$. So another major benefit is that, as opposed to STP, SLV only slightly increases feature dimensions thus making the learning and predicting more efficient.

We adopt the two normalization schemes introduced in [23] on SDV and SLV, i.e., signed square rooting and ℓ_2 normalization. As illustrated in Fig. 1, each visual word in (a) is in the end characterized by two parts, i.e., $\boldsymbol{u}_k$ in (d) and $[\boldsymbol{v}_{k_1} \dots \boldsymbol{v}_{k_G}]$ in (e). They are used to model the motion (appearance) and location cues, respectively. We summarize the outline of computing SSCV of an action sequence in Algorithm 1.

4 Experiments and Discussions

In this section, we extensively evaluate the proposed method on the two public benchmark datasets: HMDB51 [13] and YouTube [18]. In all experiments, we employ LIBLINEAR [4] as the linear SVM solver. Experimental results show that our algorithm significantly outperforms the state-of-the-art methods. Our source code for computing SSCV is available online.[1]

4.1 Experimental Setup

Datasets. The HMDB51 dataset [13] is collected from a wide range of sources from digitized movies to online videos. It contains 51 action categories and 6766 video sequences in total. This dataset includes the original videos and the stabilized version. Our evaluations are based on the original ones. We follow the same experimental setting as [13] using three training/testing splits. There are 70 videos for training and 30 videos for testing in each class. The average accuracy over the three splits is reported as the performance measurement. The YouTube dataset [18] contains 11 action classes collected under large variations in scale, viewpoint, illumination, camera motion, and cluttered background. This dataset contains 1168 video sequences in total. We follow the evaluation protocol as in [18] by using the leave-one-out cross validation for a pre-defined set of 25 groups. We report the average accuracy over all classes as the performance measurement.

Low-Level Feature Extraction. We evaluate our approach on five low-level visual contents using appearance and motion features. STIP is used to detect sparse interest points and compute HOG/HOF as the descriptor [14]. Motivated by the success of dense sampling in image classification and action recognition, we also employ the dense trajectories [26] to densely sample and track interest points from several spatial scales. Each tracked interest point generates four descriptors: HOG, HOF, trajectory (TRA), and motion boundary histogram (MBH). HOG focuses on static appearance cues, whereas HOF captures local motion information. TRA characterizes the geometric shape of a trajectory. MBH computes gradient orientation histograms from horizontal and vertical spatial derivatives of optical flow. It has been proven effective to represent motion information and suppress camera motion. So for each action sequence, we compute five features: STIP (162), HOG (96), HOF (108), TRA (30), and MBH (192), where the number in parentheses denotes the descriptor dimensionality.

[1] `http://yangxd.org/code`

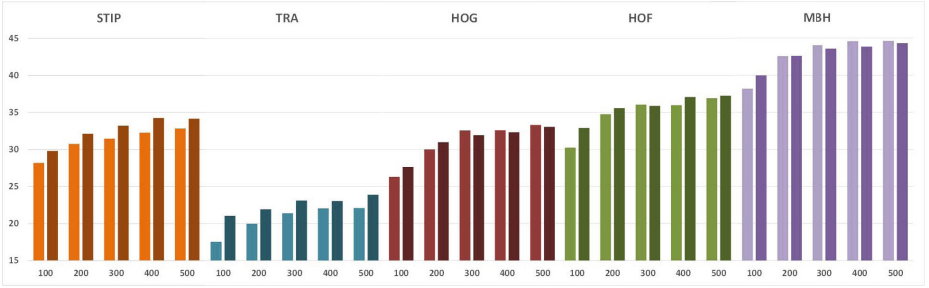

Fig. 2. Recognition accuracy (%) of FV and SDV using different descriptors with a variety of visual dictionary size K on the HMDB51 dataset. The bars in light color and deep color denote the results of FV and SDV, respectively. This figure is better viewed on screen.

4.2 Evaluation of Feature Aggregation Schemes

In this section, we compare and analyze the performance of a variety of feature aggregation schemes. We focus on the HMDB51 dataset for a detailed evaluation of the coding and pooling parameters. Note: the spatio-temporal information is discarded in the experiments of this section.

The baseline aggregation method is the hard assignment (Hard) paired with average pooling in Eq. (4-5). The local soft assignment (LocalSoft) and max pooling in Eq. (6-8) are employed with $\mathcal{K} = 10$ nearest neighbors and $\beta = 1$. We also adopt the sparse coding (SC) with max pooling in Eq. (8-9) and set the regularizer $\lambda = 1.2/\sqrt{m}$ as suggested in [20]. As a successful feature aggregation scheme, Fisher vector (FV) in Eq. (10) is compared as well. Before computing FV, we follow the preprocess in [23] [25] to apply PCA to project the descriptors to half dimensions. This step is mainly used to decorrelate the data and make it better fit the diagonal covariance matrix assumption in GMM.

SDV is compared to other feature aggregation schemes in Table 1. We set the visual dictionary size $K = 4000$ for Hard, LocalSoft, SC, and $K = 500$ for FV and SDV. As shown in this table, LocalSoft consistently outperforms Hard due to the enhanced membership estimation of descriptors to visual words. While still inferior to our method, SC largely improves the accuracy over Hard and

Table 1. Recognition accuracy (%) of different aggregation schemes using a variety of descriptors on the HMDB51 dataset

	Hard	LocalSoft	SC	FV	SDV
STIP	19.2	24.5	28.6	32.8	**34.2**
TRA	17.3	18.7	21.9	22.1	**23.9**
HOG	21.0	25.3	31.5	**33.3**	**33.1**
HOF	22.0	25.8	34.5	36.9	**37.3**
MBH	31.1	32.6	36.1	**44.6**	**44.3**

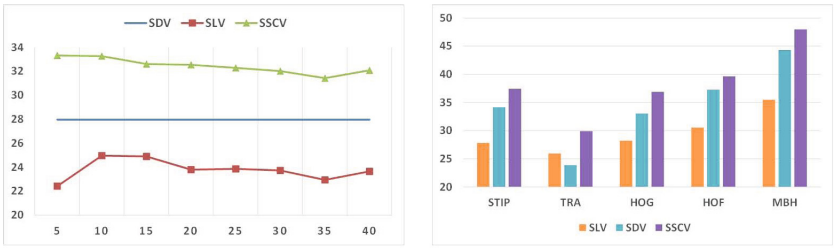

Fig. 3. Evaluations of SLV on the HMDB51 dataset. (a) Performance (%) of SLV and SSCV using STIP for a variety of visual location dictionary size G. (b) Performance (%) of SLV, SDV, and SSCV for a variety of features.

LocalSoft by introducing the sparsity in coding descriptors. SDV outperforms FV in STIP, TRA, and HOF, and yields comparable results to FV in HOG and MBH. We further conduct a detailed evaluation of FV and SDV as shown in Fig. 2. SDV systematically outperforms FV in STIP and TRA, irrespective of the visual dictionary size. For HOG, HOF, and MBH, SDV achieves higher recognition accuracy than FV in a relatively small size. SDV and FV tend to have comparable results as the visual dictionary size enlarges. In addition to the superior recognition accuracy, SDV is computationally more efficient. This is because more information is stored per visual word, which enables SDV to perform quite well by using a much more compact visual dictionary. We use $K = 500$ to compute SDV in the following experiments if not specified.

4.3 Evaluation of Spatio-temporal Models

Here we evaluate different approaches on modeling the spatio-temporal information and report results for the HMDB51 dataset.

STIP is first used to investigate the impact of the size of visual location dictionary on SLV. As shown in Fig. 3(a), the results of SLV ranges from 22.4% to 25.0% as G increases from 5 to 40. The performance of SDV is plotted as a reference. When SDV and SLV are combined to SSCV, it is not very sensitive to the size and achieves the best result using only 5 visual location words. In the following experiments, we use $G = 5$ to compute SLV. Fig. 3(b) demonstrates the results of SLV, SDV, and SSCV for a variety of features. SSCV consistently and significantly outperforms SDV for all features. This shows SLV is effective to model and provide the complementary spatio-temporal information to the motion and appearance cues in SDV. It is interesting to observe that SLV based on the pure space-time information even outperforms SDV in the feature TRA.

SSCV is then compared to the widely used spatio-temporal pyramid (STP) on modeling the space-time information. We use in our experiments four different spatio-temporal grids. For the spatial domain we employ a 1×1 whole spatial block and a 2×2 spatial grid. For the temporal domain we apply the

Table 2. Performance (%) of STP and SSCV on modeling the spatio-temporal information for a variety of features on the HMDB51 dataset

	STIP	TRA	HOG	HOF	MBH
SDV	34.2	23.9	33.1	37.3	44.3
STP	35.4 (+1.2)	28.8 (+4.9)	34.4 (+1.3)	38.1 (+0.8)	46.9 (+2.6)
SSCV	**37.4** (+3.2)	**29.9** (+6.0)	**36.9** (+3.8)	**39.7** (+2.4)	**48.0** (+3.7)

entire sequence and two temporal segments. The combination of these subdivisions in both spatial and temporal domains generate 15 space-time cells in total. We compute a separate SDV from each cell and concatenate them as the final representation of STP. As shown in Table 2, both STP and SSCV improve the results because of the spatio-temporal cues complemented to SDV. However, for all features SSCV achieves more significant improvement than STP, while with much more compact representation. In our experimental setting, the dimensions of STP and SSCV are $15mK$ and $(m+15)K$, where m is the descriptor dimensionality. So in comparison to STP, our approach can also considerably reduce the computation and memory costs in both training and testing.

Table 3. Comparison of SSCV and the state-of-the-art method for each individual feature on the HMDB51 and YouTube datasets

	HMDB51 (%)					YouTube (%)				
	STIP	TRA	HOG	HOF	MBH	STIP	TRA	HOG	HOF	MBH
WKSL'13 [26]	-	28.0	27.9	31.5	43.2	69.2	67.5	72.6	70.0	80.6
SSCV	37.4	**29.9**	**36.9**	**39.7**	**48.0**	**77.4**	**70.9**	**80.4**	**77.0**	**83.2**

4.4 Comparison to State-of-the-Art Results

In this section, we compare our results to the state-of-the-arts on the two benchmark datasets: HMDB51 and YouTube. SSCV is first compared to the results in [26] for each individual feature as demonstrated in Table 3. SSCV significantly outperforms the approach in [26], though both methods are based upon the same features. This is mainly because SDV is more representative than BOV to capture the motion and appearance information, and SLV is more effective than STP to model the spatio-temporal cues. Moreover, SSCV employs the linear SVM which is more efficient than the non-linear SVM with χ^2 kernel used in [26]. We combine all the features and compare with the most recent results in the literature as displayed in Table 4. We can observe that SSCV significantly outperforms the state-of-the-art results on the two datasets.

Table 4. Comparison of SSCV to the state-of-the-art results as reported in the cited publications

HMDB51		%	YouTube		%
GGHW'12	[6]	29.2	ICS'10	[7]	75.2
WWQ'12	[29]	31.8	LZYN'11	[17]	75.8
JDXLN'12	[11]	40.7	BSJS'11	[1]	76.5
WKSL'13	[26]	48.3	BT'10	[2]	77.8
PQPQ'13	[22]	49.2	WKSL'13	[26]	85.4
JJB'13	[9]	52.1	PQPQ'13	[22]	86.6
SSCV		**53.9**	SSCV		**88.0**

5 Conclusion

In this paper, we have presented a novel framework for action recognition. An effective coding scheme SDV is proposed to capture motion and appearance cues by sparse coding low-level descriptors and average pooling coefficient-weighted difference vectors between descriptors and visual words. A novel approach SLV is introduced to incorporate the spatio-temporal cues in a compact and discriminative manner. The combination of SDV and SLV constitutes the final representation of SSCV which jointly models the motion, appearance, and location information in a unified way. Our approach is extensively evaluated on two public benchmark datasets and compared to a number of most recent results. Experimental results demonstrate that our approach significantly outperforms the state-of-the-art methods.

Acknowledgement. This work was supported in part by NSF Grant EFRI-1137172, IIS-1400802, and FHWA Grant DTFH61-12-H-00002.

References

1. Bhattacharya, S., Sukthankar, R., Jin, R., Shah, M.: A Probabilistic Representation for Efficient Large-Scale Visual Recognition Tasks. In: CVPR (2011)
2. Brendel, W., Todorovic, S.: Activities as Time Series of Human Postures. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 721–734. Springer, Heidelberg (2010)
3. Coates, A., Ng, A.: The Importance of Encoding versus Training with Sparse Coding and Vector Quantization. In: ICML (2011)
4. Fan, R., Chang, K., Hsieh, C., Wang, X., Lin, C.: LIBLINEAR: A Library for Large Linear Classification. JMLR (2008)
5. Gemert, J., Veenman, C., Smeulders, A., Geusebroek, J.: Visual Word Ambiguity. PAMI (2009)
6. Kliper-Gross, O., Gurovich, Y., Hassner, T., Wolf, L.: Motion Interchange Patterns for Action Recognition in Unconstrained Videos. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 256–269. Springer, Heidelberg (2012)

7. Ikizler-Cinbis, N., Sclaroff, S.: Object, Scene and Actions: Combining Multiple Features for Human Action Recognition. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 494–507. Springer, Heidelberg (2010)
8. Jaakkola, T., Haussler, D.: Exploiting Generative Models in Discriminative Classifiers. In: NIPS (1998)
9. Jain, M., Jegou, H., Bouthemy, P.: Better Exploiting Motion for Better Action Recognition. In: CVPR (2013)
10. Jegou, H., Douze, M., Schmid, C., Perez, P.: Aggregating Local Descriptors into a Compact Image Representation. In: CVPR (2010)
11. Jiang, Y.-G., Dai, Q., Xue, X., Liu, W., Ngo, C.-W.: Trajectory-Based Modeling of Human Actions with Motion Reference Points. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 425–438. Springer, Heidelberg (2012)
12. Krapac, J., Verbeek, J., Jurie, F.: Modeling Spatial Layout with Fisher Vector for Image Categorization. In: ICCV (2011)
13. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: A Large Video Database for Human Motion Recognition. In: CVPR (2011)
14. Laptev, I.: On Space-Time Interest Points. IJCV (2005)
15. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning Realistic Human Actions from Movies. In: CVPR (2008)
16. Lazebnik, S., Schmid, C., Ponce, J.: Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In: CVPR (2006)
17. Le, Q., Zou, W., Yeung, S., Ng, A.: Learning Hierarchical Invariant Spatio-Temporal Features for Action Recognition with Independent Subspace Analysis. In: CVPR (2011)
18. Liu, J., Luo, J., Shah, M.: Recognizing Realistic Actions from Videos in the Wild. In: CVPR (2009)
19. Liu, L., Wang, L., Liu, X.: In Defense of Soft-Assignment Coding. In: ICCV (2011)
20. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online Dictionary Learning for Sparse Coding. In: ICML (2009)
21. McCann, S., Lowe, D.G.: Spatially Local Coding for Object Recognition. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part I. LNCS, vol. 7724, pp. 204–217. Springer, Heidelberg (2013)
22. Peng, X., Qiao, Y., Peng, Q., Qi, X.: Exploring Motion Boundary based Sampling and Spatio-Temporal Context Descriptors for Action Recognition. In: BMVC (2013)
23. Perronnin, F., Sánchez, J., Mensink, T.: Improving the Fisher Kernel for Large-Scale Image Classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010)
24. Sanchez, J., Perronnin, F., Campos, T.: Modeling the Spatial Layout of Images Beyond Spatial Pyramids. PRL (2012)
25. Sanchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image Classification with the Fisher Vector: Theory and Practice. IJCV (2013)
26. Wang, H., Klaser, A., Schmid, C., Liu, C.: Dense Trajectories and Motion Boundary Descriptors for Action Recognition. IJCV (2013)
27. Wang, H., Ullah, M., Klaser, A., Laptev, I., Schmid, C.: Evaluation of Local Spatio-Temporal Features for Action Recognition. In: BMVC (2009)
28. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-Constrained Linear Coding for Image Classification. In: CVPR (2010)

29. Wang, X., Wang, L., Qiao, Y.: A Comparative Study of Encoding, Pooling and Normalization Methods for Action Recognition. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part III. LNCS, vol. 7726, pp. 572–585. Springer, Heidelberg (2013)
30. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification. In: CVPR (2009)
31. Zhou, X., Yu, K., Zhang, T., Huang, T.S.: Image Classification Using Super-Vector Coding of Local Image Descriptors. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 141–154. Springer, Heidelberg (2010)