1-2016

A novel hierarchical Bag-of-Words model for compact action representation

Qianru SUN
Singapore Management University, qianrusun@smu.edu.sg

Qianru

Hong LIU

Hong LIU

Liqian MA

See next page for additional authors

Neurocomputing 174 (2016) 722–732

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A novel hierarchical Bag-of-Words model for compact action representation

Qianru Sun [a], Hong Liu [a,*], Liqian Ma [a], Tianwei Zhang [b]

[a] Key Laboratory of Machine Perception, Shenzhen Graduate School, Peking University, Beijing 100871, China
[b] Nakamura Lab, Department of Mechano-Informatics, School of Information Science and Technology, The University of Tokyo, Tokyo 113-8685, Japan

ARTICLE INFO

Article history:
Received 19 June 2015
Received in revised form
30 July 2015
Accepted 20 September 2015
Communicated by X. Gao
Available online 9 October 2015

Keywords:
Action representation
Bag-of-Words
Vector of Locally Aggregated Descriptors
Fisher Vectors

ABSTRACT

Bag-of-Words (BoW) histogram of local space-time features is very popular for action representation due to its high compactness and robustness. However, its discriminant ability is limited since it only depends on the occurrence statistics of local features. Alternative models such as Vector of Locally Aggregated Descriptors (VLAD) and Fisher Vectors (FV) include more information by aggregating high-dimensional residual vectors, but they suffer from the problem of high dimensionality for final representation. To solve this problem, we novelly propose to compress residual vectors into low-dimensional residual histograms by the simple but efficient BoW quantization. To compensate the information loss of this quantization, we iteratively collect higher-order residual vectors to produce high-order residual histograms. Concatenating these histograms yields a hierarchical BoW (HBoW) model which is not only compact but also informative. In experiments, the performances of HBoW are evaluated on four benchmark datasets: HMDB51, Olympic Sports, UCF Youtube and Hollywood2. Experiment results show that HBoW yields much more compact action representation than VLAD and FV, without sacrificing recognition accuracy. Comparisons with state-of-the-art works confirm its superiority further.

1. Introduction

Human action recognition has shown its significance in a large amount of applications from video surveillance to human-machine interaction [1]. Effective action representation is crucial for high recognition accuracy. Recently, successful representation models are mostly based on local space-time features such as 3D SIFT [2], HOG-HOF [3], 3D Gradients [4], and improved Dense Trajectory (iDT) [5], see [6,7] for more evaluation studies. Once local features are extracted from action videos, typically they are quantized by clustering algorithms to generate a visual codebook, then each of them is assigned to the nearest codeword. The statistic of word assignments yields the Bag-of-Words (BoW) histogram [8,9] which is very compact and robust for human action representation [6,10–13].

To improve BoW, researchers have recently developed many successful alternative models, such as Locality-constrained Linear Coding (LLC) [14], Fisher Vectors (FV) [15,16], Super Vector (SV) encoding [17], kernel codebook encoding [19], and Vector of Linearly Aggregated Descriptors (VLAD) [20,23]. Among them, VLAD and FV show outstanding performances for human action recognition [5,24,25,27–29]. Compared with BoW in Fig. 1, VLAD records the 1st-order difference between local features and codewords, i.e., the residual vectors generated by hard assignment. FV includes not only the 1st-order mean residual vectors but also the 2nd-order covariance residual vectors, which are generated by more complicated soft assignment.[1] Both of them support the potential of using residual information to get more efficient models.

Generally, model efficiency includes the time and storage costs of computing representations, learning classifiers on these representations and recognizing new videos. VLAD and FV are superior to BoW for computing representations. Taking VLAD [20] as an example, it includes high-dimensional information in each codeword, therefore when to reach a given level of performance, a small number of codewords are sufficient for VLAD. The cost of computing VLAD representation is thus greatly lower than that of BoW histogram. However, for *D-dim* local features and *K* codewords, BoW histogram is only *K-dim*, while VLAD representation is *KD-dim* because it aggregates *D-dim* residual vectors in an element-wise manner. Meanwhile, local space-time features are high-dimensional, e.g., $D = 396$ for iDT [5], so $dim_{VLAD} \gg dim_{BoW}$. High-dimensional VLAD representation results in high costs on time and storage for both training classifiers and recognizing new videos, especially on large-scale datasets like HMDB51 [39]. This

* Corresponding author.
E-mail addresses: liuh@szpku.edu.cn, hongliu@pku.edu.cn (H. Liu).

[1] Hard assignment quantizes the feature into the only codeword, while soft assignment enables the feature to be represented by multiple codewords [10].

http://dx.doi.org/10.1016/j.neucom.2015.09.074
0925-2312/

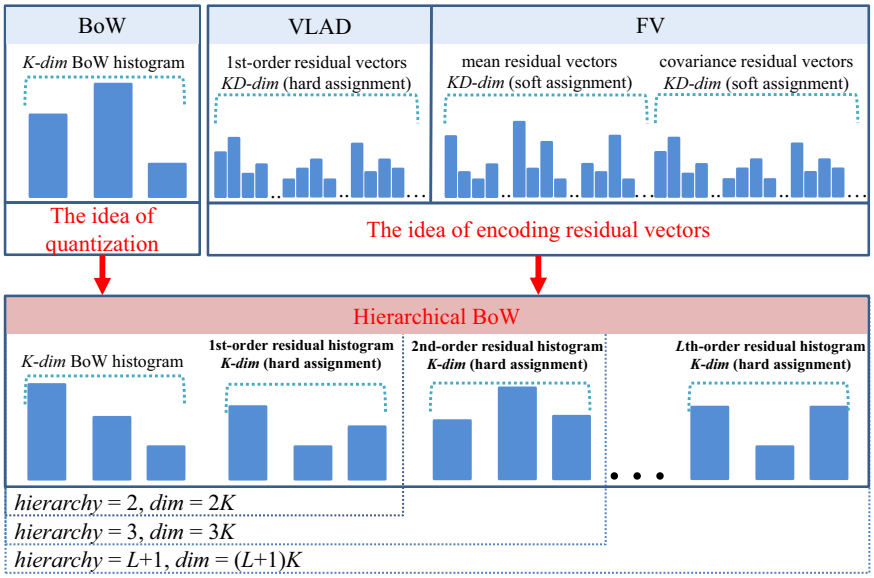

Fig. 1. The basic idea of hierarchical BoW. It is inspired by the quantization idea of BoW and the residual encoding idea of VLAD and FV. The final HBoW histogram is the concatenation of BoW histogram and multiple orders of residual histograms. K is the size of codebook and D is the dimension of local features. Hard assignment means using K-means, while soft assignment means using Gaussian Mixture Models (GMM) for probabilistic assigning weights. Note that HBoW with hierarchy=1 is same to BoW.

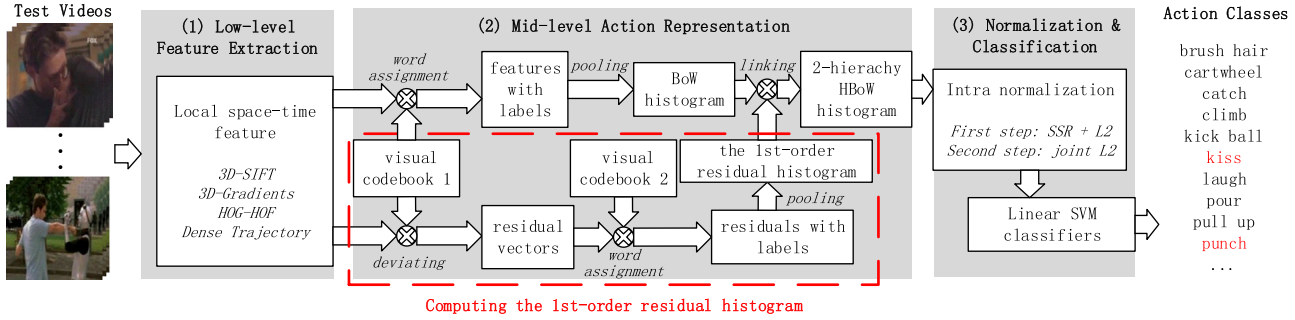

Fig. 2. The flow chart of our action recognition in videos. The 1st-order residual histogram is generated and linked to BoW histogram to form the 2-hierarchy HBoW histogram. Normalized HBoW histogram serves as the final action representation, and linear SVM classifiers are used for recognition. Note that all visual codebooks in (2) are pre-learned offline, and the normalization step in (3) is elaborated in Section 2.4.

problem is much worse for FV representation which has the dimension of $2KD$ [16,25,30].

In this paper, we show that this problem can be effectively solved by the means of BoW, for which we develop a new model called hierarchical BoW (HBoW). Its motivations are illustrated in Fig. 1. Specifically, it aims at (1) compressing residual vectors into compact residual histograms by simple but efficient BoW quantization, and (2) utilizing multiple orders of residual histograms to compensate for the quantization loss and make the final representation, called HBoW histogram, strongly informative for action recognition.

The flow chart of our action recognition method is illustrated in Fig. 2. It contains three main steps: (1) low-level feature extraction, (2) mid-level action representation, (3) normalization and classification. Our proposal of HBoW works in the second step, and an example process of generating the 1st-order residual histogram is given in the red frame of Fig. 2. The innovation lies in that the residual vectors between original local features and codewords are regarded as new features to execute word assignment again. The resulted 1st-order residual histogram is concatenated to BoW histogram to form the 2-hierarchy HBoW histogram. If we iterate this process for L times, then we get $(L+1)$-hierarchy HBoW histogram, see more details in Sections 2.2 and 2.3. If all codebooks are assumed to have K codewords, each iteration produces a K-dim residual histogram. The dimension of $(L+1)$−hierarchy HBoW histogram is therefore $K(L+1)$, which is significantly smaller than KD since $L+1 \ll D$. Then, using low-dimensional HBoW histogram for action representation saves a lot of time and storage for training classifiers and recognizing new videos.

In summary, HBoW is inherently derived from iterative BoW quantization with high-order residual vectors. It has two advantages that (1) It shares high compactness and efficiency of the original BoW; (2) It yields strongly discriminative representation by using high-order statistic information.

1.1. Related works

As we discussed, BoW model with local features has become very popular for visual understanding researches, such as image classification and object/action recognition. Alternative encoding models [14,17,18,20–22] based on BoW framework obtain the state-of-the-art performances in many visual tasks.

Wang et al. [31] and Peng et al. [10] evaluated most of these models for human action recognition, and observed that FV encoding outperforms others. FV combines the benefits of generative and discriminative approaches, and usually leverages Gaussian Mixture Model (GMM) as its dictionary. Wang et al. [5] adopted FV encoding with iDT features, and obtained generally good results on frequently-used action datasets, e.g., Olympic Sports [42], UCF Youtube [33] and

Hollywood2 [34]. Very recently, Peng et al. [29] proposed the double-layer FV to construct highly discriminative action representations. They introduced the max-margin dimensionality reduction method to compress the FV obtained from the first layer, and represented the whole video by the second-layer FV. In experiments, they achieved state-of-the-art accuracies on large-scale action datasets, e.g., HMDB51 containing 51 human actions [39].

Compared with FV, VLAD [20] shows a bit weaker performance but better time efficiency for its excellent combination with the simple but efficient *K*-means algorithm, see more evaluations in [24]. To enhance the original VLAD for action representation, Peng et al. [24] augmented it to be a high-order version of VLAD (H-VLAD) by using high-order statistic information, inspired by FV. In their experiments, H-VLAD was proved to yield better performance than FV on both large-scale image and action datasets. Our idea of encoding high-order residual information into HBoW model is in turn inspired by H-VLAD. The difference lies in that H-VLAD uses the 1st-order residual vectors to compute three-order super vectors, but HBoW generates high-order residual vectors to construct high-order residual histograms in an iterative manner. Moreover, the "order" in HBoW model could be higher than three to exploit extra complementary information while remaining high computational speed.

The rest of paper is organized as follows. Section 2 introduces the generation and normalization of HBoW histogram in detail. The complexity comparisons between HBoW and baseline BoW, VLAD and FV are presented in Section 3. Section 4 demonstrates the performances of HBoW model for action recognition on four challenging benchmark datasets [33,34,39,42], in comparison with state-of-the-art works. Conclusions are given in Section 5.

2. Proposed algorithm

In this section, the process of generating 1st-order residual histogram is firstly given on the basis of original BoW and VLAD which share the same idea of hard assignment and are both time efficient. Then, we extend this process to high-order situations and use the resulted residual histograms to form HBoW histogram. Finally, HBoW histogram is normalized in a special way for its section characteristic.

2.1. Brief reviews of BoW and VLAD

In the standard framework of BoW, a visual codebook $\mathbf{C}=[\mathbf{c}_1, \ldots, \mathbf{c}_k, \ldots, \mathbf{c}_K] \in \mathcal{R}^{D\times K}$, which corresponds to the visual codebook 1 in Fig. 2, is pre-learned by *K*-means from random training samples. Let $\mathbf{X}=[\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_N] \in \mathcal{R}^{D\times N}$ denotes the feature set of size N extracted from an action video. Each feature is assigned to the nearest codeword and gets its word label. Features assigned with the same label share the similar feature appearance. The action video is then represented by BoW histogram $\mathbf{hist}_K(\mathbf{X})$, where the k-th histogram bin is computed as follows:

$$\mathbf{hist}_k(\mathbf{X}) = \left|\{\mathbf{x}_i | LB(\mathbf{x}_i) = k\}\right| \tag{1}$$

where $LB(\mathbf{x}_i)=k$ denotes that k is the word label of $\mathbf{x}_i$, and $|.|$ counts the number. The dimension of BoW histogram equals to codebook size K.

VLAD is proposed by Jégou et al. [20]. Its codebook generation and hard assignment processes are same to those of BoW. The difference lies in that for each codeword $\mathbf{c}_k$, a vector $\mathbf{v}_k$ is derived from the element-wise aggregating of the residual vectors between $\mathbf{c}_k$ and all local features assigned to $\mathbf{c}_k$, as follows:

$$\mathbf{v}_k = \sum_{\mathbf{x}_i : LB(\mathbf{x}_i)=k} (\mathbf{x}_i - \mathbf{c}_k) \tag{2}$$

Then, the combination of all *D-dim* vectors $\mathbf{v}_k$ with $k=1,\ldots,K$ yields the VLAD representation, thus it has the dimension of *KD*.

2.2. The 1st-order residual histogram

In Eq. (2), the residual vectors $\mathbf{x}_i-\mathbf{c}_k$ are directly mapped to a *D-dim* vector by element-wise aggregating, which means the computation is between the element pair $(\mathbf{x}_{ij}, \mathbf{c}_{kj})$ with $j=1,\ldots,D$. This aggregating involves the parameter D, thus causes the problem of high dimensionality in the final encoding result. To solve this problem, we propose to compress residual vectors before encoding. The compressing method is inspired by BoW quantization, and specific steps are as follows. Firstly, we denote the bundle of residual vectors as a new feature set $\mathbf{V}^{(1)}$:

$$\mathbf{V}^{(1)} = \left\{\mathbf{v}_{ki}^{(1)} | \mathbf{v}_{ki}^{(1)} = \mathbf{x}_i - \mathbf{c}_k, i=1,\ldots,N, LB(\mathbf{x}_i)=k\right\} \tag{3}$$

where each vector $\mathbf{v}_{ki}^{(1)}$ represents the cluster-centered difference characteristic of $\mathbf{x}_i$, and the superscript denotes "1st-order". Following the BoW quantization framework, we execute *K*-means clustering on $\mathbf{V}^{(1)}$ and get the second codebook $\mathbf{C}^{(1)}=[\mathbf{c}_1^{(1)}, \ldots, \mathbf{c}_{k_1}^{(1)}, \ldots, \mathbf{c}_{K_1}^{(1)}] \in \mathcal{R}^{D\times K_1}$, which corresponds to the visual codebook 2 in Fig. 2. After word assignment from $\mathbf{V}^{(1)}$ to $\mathbf{C}^{(1)}$, each residual vector $\mathbf{v}_{ki}^{(1)}$ and its corresponding local feature $\mathbf{x}_i$ get the 1st-order residual label k_1. Features assigned to the same k_1 share similar cluster-centered difference characteristics. The global summarization of such characteristics then can be encoded in the 1st-order residual histogram $\mathbf{hist}_{K_1}(\mathbf{V}^{(1)})$, where k_1-th histogram bin is computed as

$$\mathbf{hist}_{k_1}\left(\mathbf{V}^{(1)}\right) = \left|\left\{\mathbf{v}_{ki}^{(1)} | LB(\mathbf{v}_{ki}^{(1)}) = k_1\right\}\right| \tag{4}$$

where $k_1 \in [1,\ldots,K_1]$, $i\in[1,\ldots,N]$, $k\in[1,\ldots,K]$, and $|.|$ counts the number.

Linking the 1st-order residual histogram $\mathbf{hist}_{K_1}(\mathbf{V}^{(1)})$ to original $\mathbf{hist}_K(\mathbf{X})$ yields the 2-hierarchy HBoW histogram:

$$\mathbf{Hist}^{(2)} = \left[\mathbf{hist}_K(\mathbf{X}); \mathbf{hist}_{K_1}(\mathbf{V}^{(1)})\right] \tag{5}$$

which encodes the original distribution of local features as well as their cluster-centered difference characteristics. Its superscript here denotes "2-hierarchy".

Intuitive examples of $\mathbf{hist}_K(\mathbf{X})$ and $\mathbf{hist}_{K_1}(\mathbf{V}^{(1)})$ are respectively presented in Fig. 3(a) and Fig. 3(b), (c). Note that histograms in (b and c) are in the relation of equivalence since residual vectors correspond with local features. In (b), colorful arrows represent the residual vectors from the cluster center to local features assigned to this cluster. They are actually center offset vectors, which represent the cluster-centered orientations in 2D situation. In other words, the clustering of residual vectors in (b) equals to the segmentation of angular coordinates in (c), visually illustrated as pie segmentations. Therefore, the statistic of features within different-color pies yields the same histogram.

2.3. High-order residual histograms

Similar to Eq. (3), the 2nd-order residual vectors $\mathbf{V}^{(2)}$ can be computed between vector pair $(\mathbf{v}_{ki}^{(1)}, \mathbf{c}_{k_1}^{(1)})$ in $(\mathbf{V}^{(1)}, \mathbf{C}^{(1)})$. After clustering $\mathbf{V}^{(2)}$ to $\mathbf{C}^{(2)}$ with K_2 codewords, the 2nd-order residual histogram $\mathbf{hist}_{K_2}(\mathbf{V}^{(2)})$ can be similarly computed by Eq. (4). Accordingly, it is easy to get high-order residual histograms $\mathbf{hist}_{K_3}(\mathbf{V}^{(3)}), \ldots, \mathbf{hist}_{K_L}(\mathbf{V}^{(L)})$ by iterating above process. Concatenating them to the original BoW histogram yields the $(L+1)-$hierarchy HBoW histogram:

$$\mathbf{Hist}^{(L+1)} = \left[\mathbf{hist}_K(\mathbf{X}); \ldots; \mathbf{hist}_{K_l}(\mathbf{V}^{(l)}); \ldots\right] \tag{6}$$

where $l\in[1,\ldots,L]$. For briefness, we denote $\mathbf{Hist}^{(L+1)}=[\mathbf{hist}_0; \ldots; \mathbf{hist}_l; \ldots]$. This histogram is the concatenation of one *K-dim* BoW histogram and all K_l*-dim* residual histograms, therefore its dimension is $K+\sum_{l=1}^{L} K_l$.

According to above description, the characteristic of HBoW can be summarized in three aspects: (1) its basis is the BoW quantization, hence it shares the high robustness and fast calculation velocity of

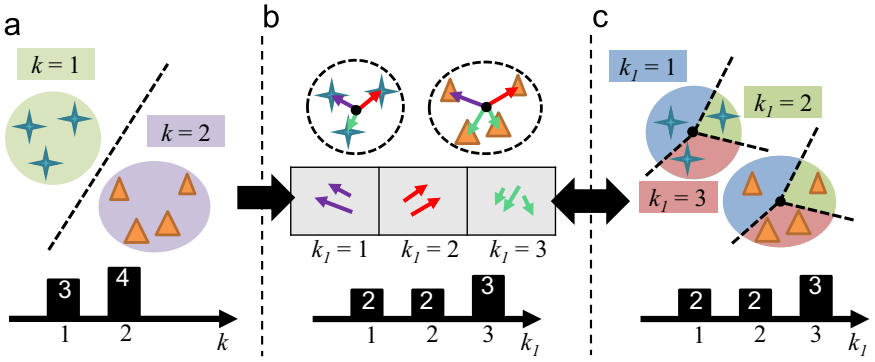

Fig. 3. Example illustrations of BoW histogram in (a), and the 1st-order residual histogram in (b) or (c). Shurikens and triangles in (a and c) represent local features, black points are cluster centers, and colorful arrows denote residual vectors. Histograms in (b and c) are equivalent since residual vectors correspond with local features.

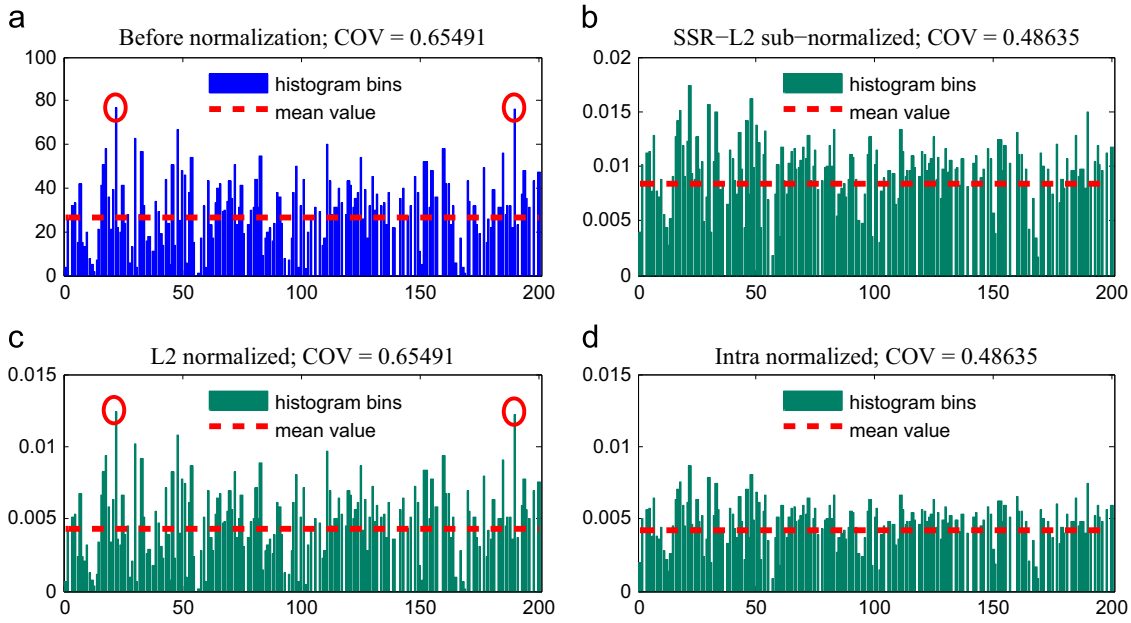

Fig. 4. The effect of different normalizing schemes for a 200-*dim* HBoW histogram. Traditional L2 normalization is from (a) to (c), where red circles indicate some obvious burst bins. Intra normalization is performed from (a) to (b), then from (b) to (d). The energy spectrum of histogram in (d) appears much more uniform. The coefficient of variation (COV) is used to measure the burstiness. Higher COV value means higher burstiness in the histogram. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

original BoW; (2) its representation is statistic histogram, thus has the advantages of histogram homogenization and scale-invariance when describing diverse samples; (3) its hierarchy is not fixed and the histogram in each hierarchy contains supplementary information.

2.4. HBoW histogram normalization

According to the standard normalizations of VLAD and FV [23,16,35,29], the representative can be enhanced by applying Signed Square Root (SSR) function (power normalization with exponent=0.5) before L2 normalization, named SSR-L2, to suppress the effect of burstiness (an effect that some histogram bins become too large compared with others, i.e., red circled bins in Fig. 4(a and c)). Since HBoW histogram is the concatenation of sub histograms, this paper adopts SSR-L2 normalization to normalize sub histograms respectively, then performs joint L2 normalization for entire histogram. This scheme is inspired by intra normalization [23] in which VLAD vector is normalized in a sectional manner. In this paper, we name it intra normalization, too. Specifically, the first step is to perform sub SSR-L2 normalization as

$$\mathbf{Hist}_{sn}^{(L+1)}=\left[\frac{sng(\mathbf{hist}_0)\cdot\mathbf{hist}_0^{0.5}}{\|\mathbf{hist}_0\|_2};\ldots;\frac{sng(\mathbf{hist}_l)\cdot\mathbf{hist}_l^{0.5}}{\|\mathbf{hist}_l\|_2};\ldots\right] \tag{7}$$

where $l=1,\ldots,L$, and *sgn* denotes sign function. The second step is joint L2 normalization formulated as

$$\mathbf{Hist}_{jn}^{(L+1)}=\frac{\mathbf{Hist}_{sn}^{(L+1)}}{\|\mathbf{Hist}_{sn}^{(L+1)}\|_2} \tag{8}$$

where the subscripts *sn* and *jn* respectively indicate sub and joint normalization.

An example of our intra normalization is illustrated in Fig. 4. The process of $\mathbf{Hist}^{(L+1)}\rightarrow\mathbf{Hist}_{sn}^{(L+1)}$ then $\mathbf{Hist}_{sn}^{(L+1)}\rightarrow\mathbf{Hist}_{jn}^{(L+1)}$ is performed with SSR-L2 normalization from (a) to (b), then joint L2 normalization from (b) to (d). The histogram obtained by traditional L2 normalization (i.e., only one step of joint L2) is presented in (c) for comparison. Obvious burst bins are labeled by red circles, and it is clear that traditional L2 normalization can not compress these bins. In contrast, intra normalized histogram in (d) shows absolutely no peaks in the energy spectrum, and bins appear more uniform. In order to compare the normalization results numerically, we use the coefficient of variation (COV), i.e., the ratio of energy spectrum standard deviation σ and energy spectrum mean μ, to represent the degree of histogram burstiness. Higher COV value means higher burstiness. COV values in (a and b) show that performing SSR-L2 normalization on sub histograms

compress COV from 0.65491 to 0.48635. Even though (a), (c) and (b), (d) show that L2 normalization have no effect on compressing COV, it is still necessary for global scale normalization.

2.5. HBoW algorithm

In Algorithm 1, we present all the computation steps of HBoW containing histogram generation and normalization. Local space-time features act as input. Original BoW histogram and the proposed residual histograms are successively generated by iterating Step 2–10. The selection of new codebook size K' in Step 12 will be discussed in Section 4.1. The normalized form of HBoW histogram obtained in Step 16 serves as the final action representation. It should be noted that in practice, the codebooks in Step 2 are usually pre-learned offline and online manipulation is only invoking corresponding codebook from storage.

3. Complexity comparison

This section compares the $(L+1)-$hierarchy HBoW with baseline models, namely BoW, VLAD, and FV, in terms of the total number of codewords (*tnc*), the complexity for assigning N local features to codewords (*ac*), and the dimensionality of action representation (*dim*), as shown in Table 1. Note that the codebook size of FV means the number of Gaussians and its soft assignment is assumed to be global assignment, i.e., each local feature is assigned with all the codewords. For HBoW, we simply assume that $K_l = K$ to give an intuitive number for comparison.

In the first row of Table 1, we observe that HBoW takes the burden of learning $K(L+1)$ codewords. However, it should not be worried about since codebooks are usually pre-learned offline in real applications. In contrast, the assignment complexity in the second row makes influence on final time efficiency, since new input features have to find their nearest codewords online. The third row shows that the dimension of HBoW histogram is rather lower than the representations of VLAD and FV, due to the fact that $L+1 \ll D < 2D$. In the next section, these models will be compared further with regard to the accuracy and computational cost in action recognition experiments.

Algorithm 1. Computing HBoW histogram for the action video.

Input: Local space-time features $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_i, ..., \mathbf{x}_N] \in \mathcal{R}$ extracted in the video, initial codebook size K, and hierarchy parameter L.

Output: $(L+1)$-hierarchy HBoW histogram **Hist**.

1 **for** $l = 1$ *to* $L+1$ **do**

2 Compute the visual codebook $\mathbf{C} = [\mathbf{c}_1, ..., \mathbf{c}_k, ..., \mathbf{c}_K]$ using K-means on the random subset of $\mathbf{X}$. % This step is usually finished offline with training videos.

3 **for** $i = 1$ *to* N **do**

4 For $\mathbf{x}_i$, find its nearest codeword $\mathbf{c}_k$ in $\mathbf{C}$ with

$$k \leftarrow \arg\min_j \|\mathbf{x}_i - \mathbf{c}_j\|, j = 1, ..., K$$

5 Assign $\mathbf{x}_i$ with

$$LB(\mathbf{x}_i) \leftarrow k.$$

6 Update the k-th bin of histogram $\mathbf{hist}_K(\mathbf{X})$ with

$$\mathbf{hist}_k(\mathbf{X}) \leftarrow \mathbf{hist}_k(\mathbf{X}) + 1.$$

7 Compute residual vector

$$\mathbf{v}_{ki} \leftarrow \mathbf{x}_i - \mathbf{c}_k.$$

8 Store $\mathbf{v}_{ki}$ in $\mathbf{V}$.

9 **end**

10 Normalize $\mathbf{hist}_K(\mathbf{X})$ with

$$\mathbf{hist}_K(\mathbf{X}) \leftarrow \frac{sgn[\mathbf{hist}_K(\mathbf{X})] \cdot [\mathbf{hist}_K(\mathbf{X})]^{0.5}}{||\mathbf{hist}_K(\mathbf{X})||_2}.$$

11 Concatenate $\mathbf{hist}_K(\mathbf{X})$ to **Hist**, then set $\mathbf{hist}_K(\mathbf{X})$ to null.

12 Select the new K'.

13 Substitute K' for K.

14 Substitute $\mathbf{V}$ for $\mathbf{X}$.

15 **end**

16 Normalize **Hist** with

$$\mathbf{Hist} \leftarrow \frac{\mathbf{Hist}}{||\mathbf{Hist}||_2}.$$

17 **return Hist**.

Table 1
Complexity and dimensionality comparisons among BoW, VLAD, FV, and HBoW.

Parameter	BoW	VLAD	FV	HBoW
tnc	K	K	K	$K(L+1)$
ac	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(NK)$	$\mathcal{O}(NL)$
dim	K	DK	$2DK$	$K(L+1)$

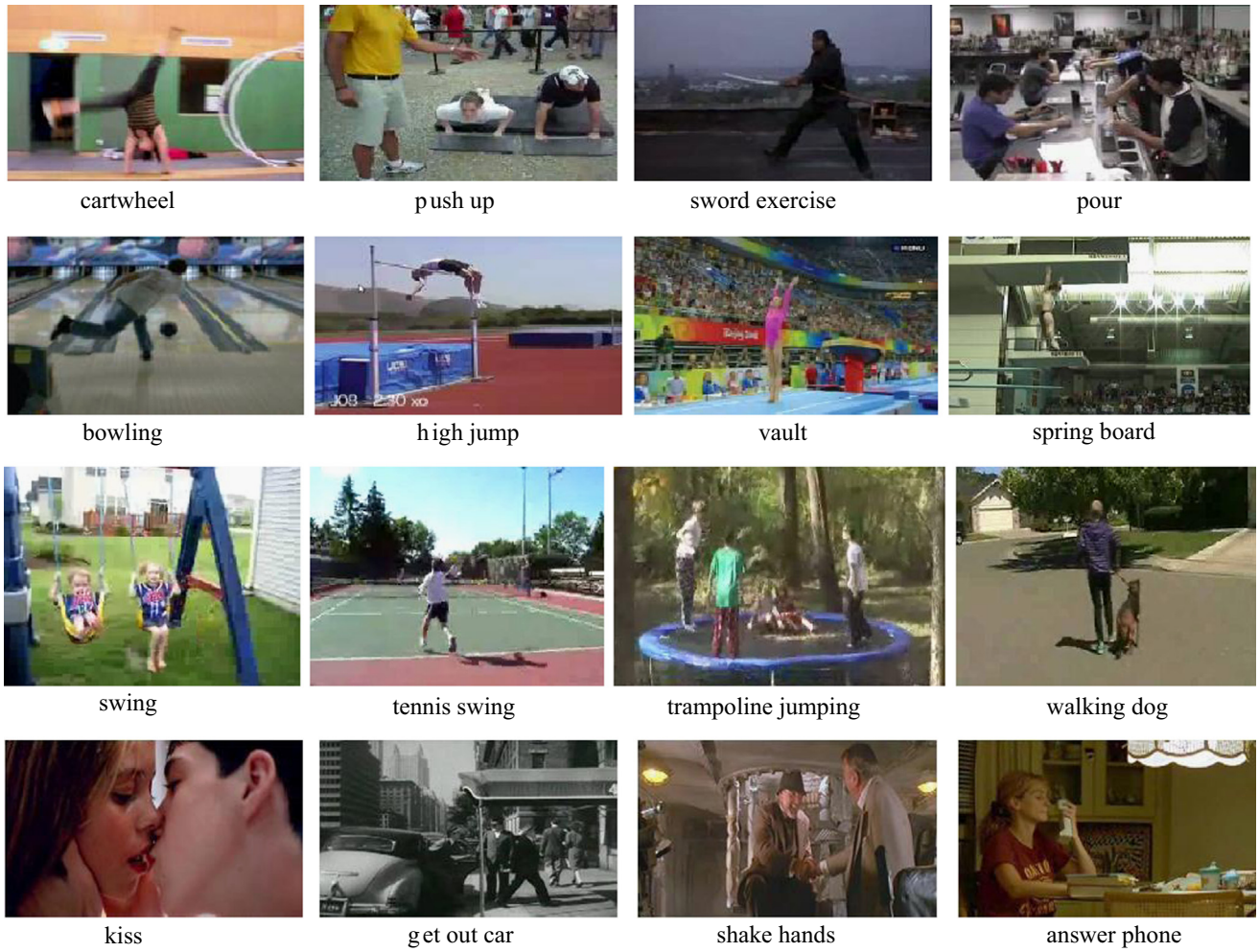

Fig. 5. Sample frames from action datasets used in our experiments. From top to bottom: HMDB51, Olympic Sports, UCF Youtube, and Hollywood2.

4. Experimental validation

In this section, the experiments of action recognition are carried out on four benchmark datasets: HMDB51 containing 6766 videos of 51 human actions collected from various sources [39], Olympic Sports containing 783 videos of 16 sport actions [42], UCF Youtube containing 1168 videos of 11 human actions from Youtube [33], and Hollywood2 containing 1707 videos depicting 12 actions collected from 69 different Hollywood movies [34]. Sample frames of these datasets are given in Fig. 5.

On HMDB51, we follow the standard evaluation protocol in [39], and report the average classification accuracy over three test-train splits that per category contains 70 examples for training, and 30 for testing. On Olympic Sports, we use the provided train/test split in [42] as 17 to 56 training samples and 4 to 11 test samples per class. The recognition performance is measured in terms of mean average precision (mAP) over all action classes. The mAP measurement is also used for Hollywood2 dataset, where the standard training-testing splits are 823 videos for training and 884 videos for performance testing [34]. Training and testing video clips come from different movies. On UCF Youtube, we use Leave-One-Out Cross-Validation for the pre-defined setting of 25 groups in [33]. Average accuracy over all classes is reported as the performance measure.

All experiments are carried out on an Intel Core i5-3470 CPU with 3.20 GHz. Our software relies on Microsoft Visual Studio 2012 and Matlab 2012a. Baseline BoW (L2 normalized), VLAD (intra normalized [23]) and FV (SSR-L2 normalized [16]) and their dictionary learning algorithms, namely K-means and GMM, are implemented with VLFeat Toolbox [38]. The widely used 162-*dim* HOG-HOF [43] and 396-*dim* iDT (the combination of HOG, HOF, and MBH) [5] are adopted as two kinds of local space-time features, namely sparse features and dense features. To learn visual codebooks, each iteration we randomly extract a 1/3 subset of local features (or residual vectors) as training samples. In recognition

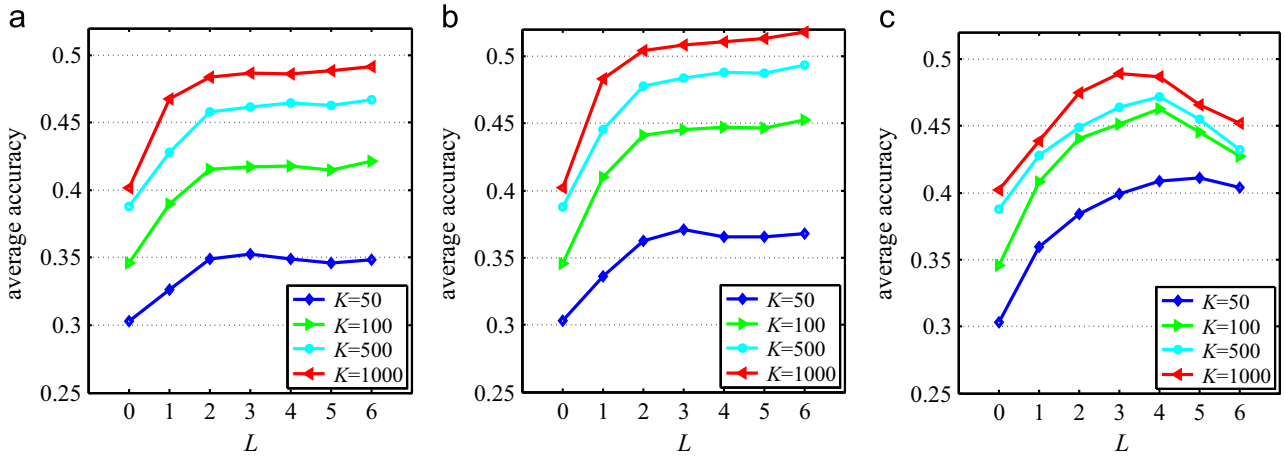

Fig. 6. Recognition accuracies on HMDB51 with respect to the initial codebook size K and hierarchy parameter L.

steps, we train linear SVM classifiers for all models except BoW.[2] All reported results are the average accuracy of 10 runs for existing randomness, e.g., in the initialization of K-means.

4.1. Parameter tests

The setting of codebook size K is important for clustering. Since HBoW has multiple clustering steps, the problem of selecting K, K_l becomes much pressing. Related works determine K by testing discrete candidates [35,5]. We follow the recognition setup of [35], and set initial K (i.e., K_0) ranging in $[50, 100, 500, 1000]$. Assuming K_l is determined, we use a rough scale factor $\rho = \{\frac{1}{2}, 1, 2\}$ to control K_{l+1}, allowing $K_{l+1} = \rho K_l$. The hierarchy parameter L is set from 0 to 6, where HBoW with $L=0$ is same to original BoW.

The average recognition accuracies on HMDB51 with HOG-HOF features are given in Fig. 6. It is easy to observe that larger initial K always brings better results, which suits well to the performance characteristic of BoW based models. With regard to hierarchy L, most curves are stably increasing with L, indicating that the residual histogram in each hierarchy contains supplementary information for action representation. However, when $L > 3$, some of them appear slight increasing even decreasing tendencies. A possible reason is that codebook learning error accumulates too much during more and more iterative clusterings on random training features. If the training feature set is assumed to be large enough, the obtained codebook could be more stable to depict the feature space, and the decreasing tendencies could be weakened.

In particular, the curve decreasing appears much obvious at $\rho = 2$, because the over-segmentation of residual vector space are severe when codebook size K_l grows exponentially. Across (a–c), we notice that the curves of $K = 50$ keep growing from $\rho = 0.5$ to 2. It is due to the fact that the distribution of local features can not be fully modeled when initial K is too small, and using larger K_l in subsequent clusterings can dig out more discriminative information from the residual vector space. Finally, the highest accuracy 51.76% is achieved at $K = 1000, \rho = 1, L = 6$. In following experiments, $K = 1000, \rho = 1, L = 6$ are used by default.

4.2. Normalization tests

In Fig. 7, we show the average recognition accuracies or mAPs using different normalization schemes for HBoW histogram. Respectively, global L2 is the traditional L2 normalization on the entire histogram, global SSR-L2 is performing SSR-L2 directly on the entire histogram, sub SSR-L2 is only the first step of our intra normalization without any global normalizer, and sub SSR-L2 + joint L2 is exactly the proposed intra normalization in Eqs. (7) and (8).

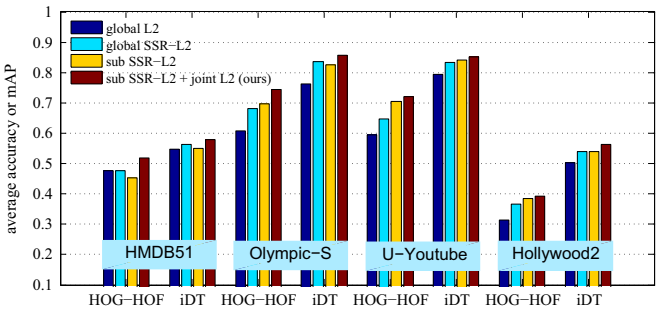

Fig. 7. The recognition results on four datasets using HBoW with different normalization schemes.

From Fig. 7, we can come to following conclusions. (1) Our proposal of intra normalization greatly improves the model efficiency, in comparison with traditional L2 or the standard SSR-L2 which has been successfully used for the normalization of VLAD and FV. This is due to HBoW histogram has the special structure that each entire histogram is composed by sub histograms derived from discriminative residual spaces. Sectional normalization helps to compress the scale-varying burstiness in different sub histograms (from different residual spaces). (2) Our intra normalization brings more improvements for sparse HOG-HOF than dense iDT on most datasets. The possible reason is that the histogram of sparse local features tends to be sparser, which indirectly increases the influence of burstiness. Our intra normalization can compress such burstiness effectively, thereby making normalized histogram much better for action representation.

4.3. Comparison with baselines

This section compares HBoW with BoW, VLAD and FV firstly using 162-*dim* HOG-HOF features on HMDB51 dataset. Fig. 8 shows the recognition accuracy in (a and b) and average recognition time in (c and d) respectively as the function of both codebook size K and representation dimension *dim*.

In Fig. 8(a), it is obvious that given a fixed-size codebook, other models significantly outperform BoW (with linear and nonlinear SVMs). This is not surprising for VLAD and FV, since they include extremely higher dimensional information than BoW histogram, e.g., when $K=500$, $dim_{BoW}=500$, but $dim_{VLAD}=81{,}000$ and

[2] Since BoW model can achieve far superior performance by using RBF-χ^2 kernel function [6,26], the results of BoW with nonlinear SVM (named BoW-non) are presented in Fig. 8 for comparison.

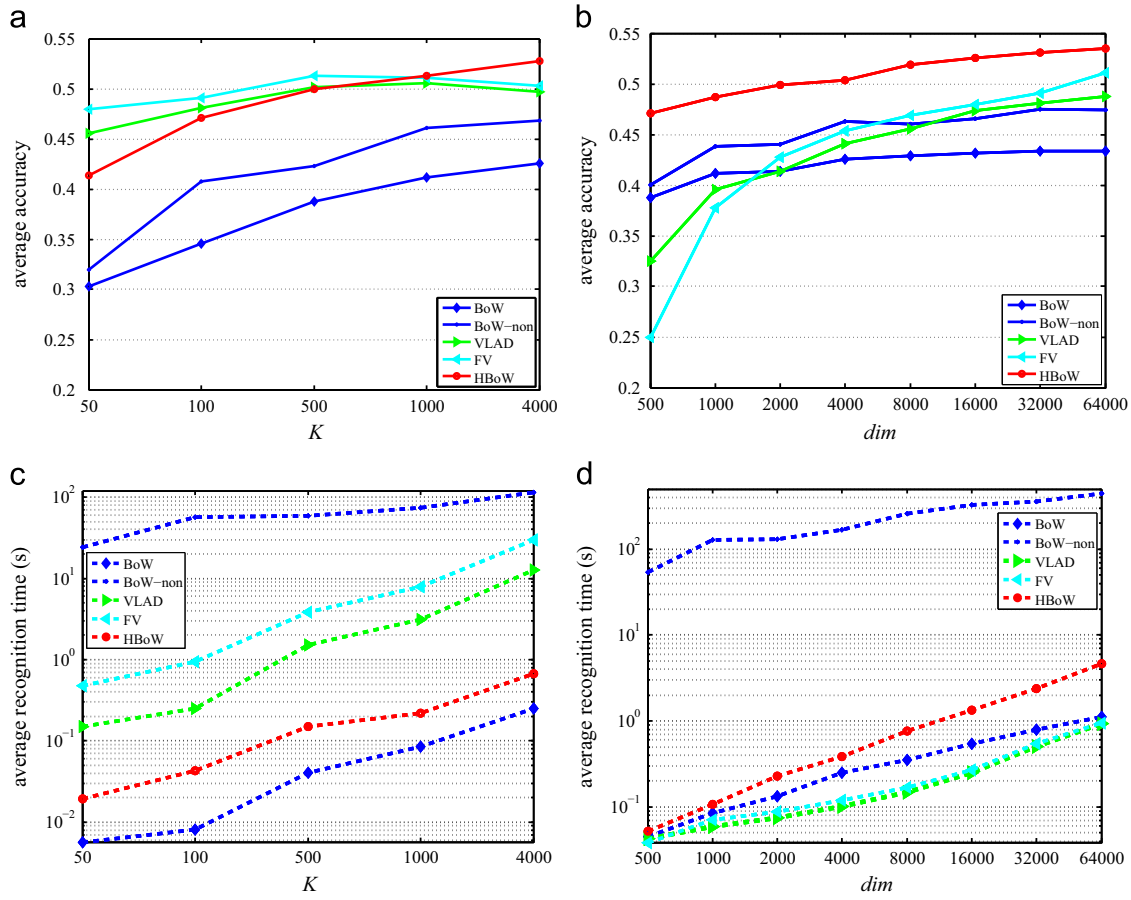

Fig. 8. Recognition accuracies and time costs as the function of codebook size K and representation dimension *dim*, using 162-*dim* HOG-HOF features on HMDB51. Note that "BoW-non" indicates BoW + nonlinear SVMs (RBF-χ^2 kernel).

Table 2
Initial codebook size K for other three datasets, referred to the conclusions of [10].

Model	BoW	VLAD	FV	HBoW
codebook size K	8000	256	256	1000

dim_{FV}=162,000. Under the same situation, our HBoW histogram is only 3500-*dim*, but its performance is very comparable to VLAD and FV. More importantly, as shown in Fig. 8(c), the average recognition speed of HBoW is rather fast, e.g., about 30 times faster than FV at K=4000. In Fig. 8(b), the accuracy superiority of HBoW becomes much more obvious, when representation dimensions are assumed to be same. This validates the strong information digging ability of HBoW, i.e., using limited dimensions to encode more distinctiveness of human action patterns.

For other three datasets, we collect recognition data using different codebook sizes for different encoding models, shown in Table 2. All codebook sizes are referred to the conclusions of an action recognition survey article [10] that *for a good balance between performance and efficiency, sizes of 256 and 8000 are good choices for super vector based encoding and other encoding respectively*. The experiment results based on these settings are given in Table 3 For comparisons with fair parameters, the results with same cluster num K=1000 are additionally presented in the right half table. Due to the poor performance of BoW with linear SVMs, we just present the results of BoW with nonlinear SVMs but others with linear SVMs. In Table 3, we observe that BoW with nonlinear SVMs results satisfying performances but suffers from extremely high computational costs. Compared with FV and VLAD, HBoW slightly outperforms in terms of recognition rates. However, the advantages of HBoW mostly manifest in low dimensionality as well as its contribution to fast computational speed, see *dim* and time rows of Table 3.

Table 3 shows that iDT outperforms HOG-HOF on all datasets. This is due to the fact that iDT combining HOG, HOF, and MBH, benefit from the complementary information encoded in each of them. Moreover, iDT is the improved version of Dense Trajectory (DT) [46]. It benefits from the additional human body detection and preserves foreground body features for effective representation [5]. We also notice that the difference between FV and (VLAD, HBoW) for iDT seems smaller than that for HOG-HOF, which is similar to the conclusion in [10]. The possible reason is that denser features can train more stable codebooks for hard assignment models, namely VLAD and HBoW, and can indirectly decrease the influence of soft assignment on FV. Besides, the information contained in the 2nd-order covariance statistics may be less complementary to the 1st-order mean statistics for iDT. Finally, for super vector based models FV and VLAD, using bigger codebook size K=1000 does not always brings better performances than K=256, which is however totally positive for BoW model, see the results of K=8000, 1000 for BoW.

4.4. Comparison with state-of-the-arts

Since best performances are always obtained when encoding extra geometric information [5,35], this paper adopts the widely used spatial-temporal pyramids (STP) model [37]. Videos are divided into two temporal parts, and three spatial horizontal parts, based on the

Table 3
Performance comparisons on Olympic Sports, UCF Youtube and Hollywood2. The "rate" denotes mAP for Olympic Sports and Hollywood2, and average accuracy for UCF Youtube. "BoW-non" indicates BoW + nonlinear SVMs (RBF-χ^2 kernel).

Olympic-S		with K in Table 2				with $K=1000$		
		BoW-non	VLAD	FV	HBoW	BoW-non	VLAD	FV
HOG-HOF	rate (%)	64.12	69.90	72.53	**74.45**	61.33	67.47	72.93
	dim	8000	41,472	82,944	7000	1000	162,000	324,000
	time (*s*)	128.30	0.72	2.31	0.09	57.83	2.45	9.70
iDT	rate (%)	81.64	84.33	85.46	**85.80**	77.60	84.72	85.00
	dim	8000	101,376	202,752	7000	1000	386,000	792,000
	time (s)	130.14	1.77	2.56	0.73	67.59	7.32	10.05
U-Youtube		with K in Table 2				with $K=1000$		
		BoW-non	VLAD	FV	HBoW	BoW-non	VLAD	FV
HOG-HOF	rate (%)	64.76	69.40	**72.52**	72.09	55.03	68.56	72.06
	dim	8000	41,472	82,944	7000	1000	162,000	324,000
	time (s)	100.39	1.21	4.57	0.21	46.77	6.32	18.30
iDT	rate (%)	76.99	83.44	84.07	**85.14**	74.10	83.71	84.66
	dim	8000	101,376	202,752	7000	1000	386,000	792,000
	time (s)	124.05	1.89	2.85	0.17	53.25	6.78	13.14
Hollywood2		with K in Table 2				with $K=1000$		
		BoW-non	VLAD	FV	HBoW	BoW-non	VLAD	FV
HOG-HOF	rate (%)	37.22	36.29	39.50	41.13	34.46	39.45	**41.53**
	dim	8000	41,472	82,944	7000	1000	162,000	324,000
	time (s)	69.05	1.03	2.47	0.43	47.55	5.30	8.09
iDT	rate (%)	49.50	53.95	55.80	**56.30**	46.02	55.93	56.11
	dim	8000	101,376	202,752	7000	1000	386,000	792,000
	time (s)	152.00	2.51	3.96	0.91	74.36	7.36	12.18

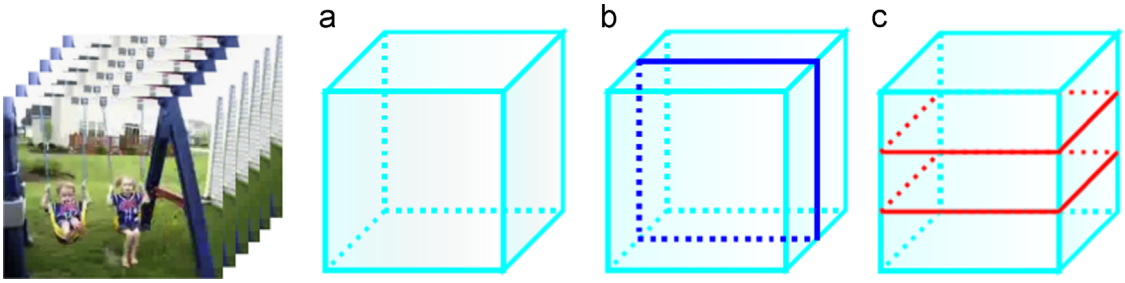

Fig. 9. An sample video is visualized as 3D cubes, referred to the STP modes in [5]. Grids in (a–c) represent the spatial-temporal pyramid modes used in our experiments.

STP used in [35]. Totally, six HBoW histograms: one for the whole video (Fig. 9(a)), two for temporal parts (Fig. 9(b)), and three for spatial parts (Fig. 9(c)) are concatenated to represent the video.

In the upper half of Table 4, HBoW with STP shows superior performance over that without STP. For instance, when using iDT features, HBoW+STP brings about 5.34%, 5.63%, 9.36% and 6.36% improvements over HBoW on four datasets, respectively. This validates the complementary efficiency of STP for our global representation model – HBoW. Besides, we believe that more complex STP modes can be employed to get much better performances as long as they do not bring about too much computation.

In the lower half of Table 4, we present the state-of-the-art results on all datasets. Oneata et al. [35] get the highest accuracy on the challenging Hollywood2 by selecting local features and encoding methods carefully. Taralova et al. [28] encode non-coarse supervoxels to BoW codewords, and obtain good results on HMDB51 and UCF Youtube. Iosifidis et al. [48] propose the kernel formulation of graph embedded extreme learning machine (GEKELM) to resolve classification problems, and achieve the best results on Olympic Sports. Compared with these works, our HBoW integrated with STP modes achieves better results by combining geometric information in video space with discriminative high-order residual information in feature space. Moreover, our result on HMDB51 is very comparable to the state-of-the-art accuracy reported in [29]. Peng et al. [29] use the double-layer FV to construct discriminative action representations. Though they propose to reduce the dimension of first-layer FV descriptors, their final action representation still follows the standard protocol of FV encoding, which is very high-dimensional if there is no additional principal component analysis (PCA) and Whitening for dimension reduction. In contrast, our HBoW histogram could be more compact since it inherently compress high-dimensional residual vectors for all clustering layers.

5. Conclusions

To get compact and efficient action representation, this paper proposes a hierarchical BoW (HBoW) model which can compress high-dimensional residual vectors into low-dimensional residual histograms in an iterative manner. Concatenating these histograms yields the action representation called HBoW histogram which is much more compact than original VLAD and FV. Therefore, as video features and

Table 4
Comparing the proposed HBoW with state-of-the-art models on HMDB51, Olympic Sports, UCF Youtube and Hollywood2 datasets.

Method	Local feature	HMDB51	Olympic-S	U-Youtube	Hollywood2
HBoW	HOG-HOF	51.76	74.45	72.09	46.63
	iDT	57.95	85.80	85.14	58.55
HBoW+STP	HOG-HOF	54.15	80.37	77.05	50.90
	iDT	**63.29**	**91.43**	**94.50**	**64.91**
Jiang et al. [40]	TrajMF+DT	40.7	80.6	–	59.5
Jain et al. [41]	DT+DCS	52.1	83.2	–	62.5
Wang et al. [5]	iDT	48.3	77.2	85.4	59.9
Sun and Liu [44]	HOG-HOF	44.3	84.5	80.3	48.66
Oneata et al. [35]	MBH+SIFT	54.8	84.6	89.0	**63.3**
Iosifidis et al. [36]	HOG-HOF	–	–	–	48.5
Taralova et al. [28]	iDT	58.8	–	88.9	–
Cai and Qiao [45]	iDT	55.9	–	–	–
Peng et al. [29]	iDT	**66.79**	–	**93.77**	–
Iosifidis et al. [47]	iDT	–	88.89	–	61.69
Iosifidis et al. [48]	iDT	–	**89.74**	–	62.5

datasets are likely to increase in size, our compact model will become more and more influential on promoting the recognition speed. Moreover, as our model can encode the high-order statistic information of features, it performs strong comparability to the state-of-the-art models for recognizing the action videos of challenging datasets.

HBoW makes use of residual vectors, therefore two kinds of frameworks are available according to the assumption that residual vectors are from non-probabilistic hard assignment, like in VLAD; or from probabilistic soft assignment, like in FV. In this paper, we adopt the former one for modeling, and our future work is to extend our idea with soft assignment which is more complicated but is expected to get more discriminative ability.

Acknowledgments

This work is supported by National Natural Science Foundation of China (NSFC, No. 61340046), National High Technology Research and Development Program of China (863 Program, No. 2006AA04Z247), Scientific and Technical Innovation Commission of Shenzhen Municipality (No. JCYJ20120614152234873, JCYJ20130331144716089), Specialized Research Fund for the Doctoral Program of Higher Education (No. 20130001110011).

References

[1] R. Marfil, J. Dias, F. Escolano, Recognition and action for scene understanding, Neurocomputing 161 (2015) 1–2.
[2] P. Scovanner, S. Ali, M. Shah, A 3-dimensional sift descriptor and its application to action recognition, in: ACM International Conference on Multimedia, 2007, pp. 357–360.
[3] I. Laptev, M. Marszalek, C. Schmid, Learning realistic human actions from movies, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[4] A. Kläser, M. Marszalek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, in: British Machine Vision Conference, 2008, pp. 995–1004.
[5] H. Wang H, A. Kläser, C. Schmid, C.L. Liu, Dense trajectories and motion boundary descriptors for action recognition, Int. J. Comput. Vis. 103 (1) (2013) 60–79.
[6] H. Wang, M.M. Ullah, A. Kläser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, in: British Machine Vision Conference, 2009, pp. 124.1–124.11.
[7] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, The devil is in the details: an evaluation of recent feature encoding methods, in: British Machine Vision Conference, 2011, pp. 1–12.
[8] J. Malik, P. Perona, Preattentive texture discrimination with early vision mechanisms, J. Opt. Soc. Am. A 7 (5) (1990) 923–932.
[9] T. Leung, J. Malik, Recognizing surfaces using three-dimensional textons, in: IEEE International Conference on Computer Vision, vol. 2, 1999, pp. 1010–1017.
[10] X. Peng, L. Wang, X. Wang, Y. Qiao, Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. arXiv preprint, 2014, arXiv:1405.4506.
[11] Y. Tian, Q. Ruan, G. An, W. Xu, Context and locality constrained linear coding for human action recognition, Neurocomputing, 2015, in press.
[12] M. Liu, H. Liu, Q. Sun, Action classification by exploring directional co-occurrence of weighted STIPS, in: IEEE International Conference on Image Processing, 2014, pp. 1460–1464.
[13] J. Yu, M. Jeon, W. Pedrycz, Weighted feature trajectories and concatenated bag-of-features for action recognition, Neurocomputing 131 (5) (2014) 200–207.
[14] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3360–3367.
[15] F. Perronnin, C. Dance, Fisher kernels on visual vocabularies for image categorization, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[16] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, Lecture Notes in Computer Science, vol. 6314, 2010, pp. 143–156.
[17] X. Zhou, K. Yu, T. Zhang, T.S. Huang, Image classification using super-vector coding of local image descriptors, Lecture Notes in Computer Science, vol. 6315, 2010, pp. 141–154.
[18] X. Lian, Z. Li, B. Lu, L. Zhang, Max-margin dictionary learning for multiclass image categorization, Lecture Notes in Computer Science, vol. 6314, 2010, pp. 157–170.
[19] J.C. van Gemert, J.M. Geusebroek, C.J. Veenman, A.W.M. Smeulders, Kernel codebooks for scene categorization, Lecture Notes in Computer Science, vol. 5304, 2008, pp. 696–709.
[20] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, C. Schmid, Aggregating local image descriptors into compact codes, IEEE Trans. Pattern Anal. Mach. Intell. 43 (9) (2012) 1704–1716.
[21] H. Liu, M. Yuan, F. Sun., RGB-D action recognition using linear coding, Neurocomputing 149 (2015) 79–85.
[22] F. Moayedi, Z. Azimifar, R. Boostani, Structured sparse representation for human action recognition, Neurocomputing 161 (2015) 38–46.
[23] R. Arandjelović, A. Zisserman, All about VLAD, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 1578–1585.
[24] X. Peng, L. Wang, Y. Qiao, Q. Peng, Boosting VLAD with supervised dictionary learning and high-order statistics, Lecture Notes in Computer Science, vol. 8691, 2014, pp. 660–674.
[25] X. Peng, Y. Qiao, Q. Peng, Q. Wang., Large margin dimensionality reduction for action similarity labeling, Signal Process. Lett. 21 (8) (2014) 1022–1025.
[26] J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: a comprehensive study, Int. J. Comput. Vis. 73 (2) (2007) 213–238.
[27] D. Oneata, J. Verbeek, C. Schmid, Efficient action localization with approximately normalized Fisher vectors, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2545–2552.
[28] E.H. Taralova, Fernando De La Torre, M. Hebert, Motion words for videos, Lecture Notes in Computer Science, vol. 8689, 2014, pp. 725–740.
[29] X. Peng, C. Zou, Y. Qiao, Q. Wang, Action recognition with stacked fisher vectors, Lecture Notes in Computer Science, vol. 8693, 2014, pp. 581–595.
[30] S. Ozkan, T. Ates, E. Tola, M. Soysal, E. Esen, Performance analysis of state-of-the-art representation methods for geographical image retrieval and categorization, IEEE Geosci. Remote Sens. Lett. 11 (11) (2014) 1996–2000.
[31] X. Wang, L. Wang, Y. Qiao, A comparative study of encoding, pooling and normalization methods for action recognition, Lecture Notes in Computer Science, 2012, pp. 572–585.
[33] J. Liu, J. Luo, M. Shah, Recognizing realistic actions from videos, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1996–2003.
[34] M. Marszalek, I. Laptev, C. Schmid. Actions in context, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2929–2936.
[35] D. Oneata, J. Verbeek, C. Schmid, Action and event recognition with Fisher vectors on a compact feature set, in: IEEE International Conference on Computer Vision, 2013, pp. 1817–1824.
[36] A. Iosifidis, A. Tefas, I. Pitas, Discriminant bag of Words based representation for human action recognition, Pattern Recognit. Lett. 49 (2014) 185–1924.
[37] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 2169–2178.
[38] A. Vedaldi, B. Fulkerson, VLFeat: an open and portable library of computer vision algorithms ⟨http://www.vlfeat.org/⟩, 2008.
[39] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre. HMDB: a large video database for human motion recognition, in: IEEE International Conference on Computer Vision, 2011, pp. 2556–2563.
[40] Y.G. Jiang, Q. Dai, X. Xue, W. Liu, C.W. Ngo, Trajectory-based modeling of human actions with motion reference points, Lecture Notes in Computer Science, vol. 7576, 2012, pp. 425–438.
[41] M. Jain, H. Jégou, P. Bouthemy, Better exploiting motion for better action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2555–2562.
[42] J.C. Niebles, C.W. Chen, F.F. Li, Modeling temporal structure of decomposable motion segments for activity classification, Lecture Notes in Computer Science, vol. 6312, 2010, pp. 392–405.
[43] I. Laptev, Spatio-temporal interest point library, 2011. ⟨www.di.ens.fr/laptev/~interestpoints.html⟩.

[44] Q. Sun, H. Liu, Inferring ongoing human activities based on recurrent self-organizing map trajectory, in: British Machine Vision Conference, 2013, pp. 11.1–11.10.
[45] Z. Cai, Y. Qiao, Multi-view super vector for action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 596–603.
[46] H. Wang, A. Kläser, C. Schmid, C. Liu, Action Recognition by dense trajectories, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 3169–3176.
[47] A. Iosifidis, A. Tefas, I. Pitas, Class-specific reference discriminant analysis with application in human behavior analysis, IEEE Trans. Hum.-Mach. Syst. 45 (2015) 315–326.
[48] A. Iosifidis, A. Tefas, I. Pitas, Graph embedded extreme learning machine, IEEE Trans. Cybern. (99) (2015), http://dx.doi.org/10.1109/TCYB.2015.2401973.

Qianru Sun received the Bachelor Degree of Information Science and Technology in 2010, and is working toward the Doctor Degree in the School of EE&CS, Peking University (PKU), China.

Her research interests include human action recognition & anomaly detection. She has published articles in British Machine Vision Conference (BMVC), Asian Conference on Computer Vision (ACCV), IEEE International Conference on Image Processing (ICIP) and IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

Hong Liu received the Ph.D. degree in mechanical electronics and automation in 1996, and serves as a Full Professor in the School of EE&CS, Peking University (PKU), China. Liu has been selected as Chinese Innovation Leading Talent supported by "National High-level Talents Special Support Plan" since 2013.

He is also the Director of Open Lab on Human Robot Interaction, PKU, his research fields include computer vision and robotics, image processing, and pattern recognition. Liu has published more than 150 papers and gained Chinese National Aero-space Award, Wu Wenjun Award on Artificial Intelligence, Excellence Teaching Award, and Candidates of Top Ten Outstanding Professors in PKU. He is an IEEE member, vice president of Chinese Association for Artificial Intelligent (CAAI), and vice chair of Intelligent Robotics Society of CAAI. He has served as keynote speakers, co-chairs, session chairs, or PC members of many important international conferences, such as IEEE/RSJ IROS, IEEE ROBIO, IEEE SMC and IIHMSP, recently also serves as reviewers for many international journals such as Pattern Recognition, IEEE Transactions on Signal Processing, and IEEE Transactions on PAMI.

Liqian Ma received the Bachelor Degree of Electronic Science and Technology in 2013, and is working toward the Master Degree in the School of EE&CS, Peking University (PKU), China.

His research interests include action recognition and localization. He has published articles in IEEE International Conference on Image Processing (ICIP), IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) and IEEE Signal Processing Letters.

Tianwei Zhang received his Master Degree of Electronic Science and Technology in 2013, and is working toward the Doctor Degree in the Department of Mechanoinformatics, The University of Tokyo, Japan.

His research interests include robotic motion planning, video processing and 3-D reconstruction. He has published several articles in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) and International Conference on Robotics and Biomimetics (ROBIO).