# Action and Event Recognition with Fisher Vectors on a Compact Feature Set

Dan Oneaţă Jakob Verbeek Cordelia Schmid

LEAR, INRIA Grenoble – Rhône-Alpes, France Laboratoire Jean Kuntzmann

firstname.lastname@inria.fr

## Abstract

*Action recognition in uncontrolled video is an important and challenging computer vision problem. Recent progress in this area is due to new local features and models that capture spatio-temporal structure between local features, or human-object interactions. Instead of working towards more complex models, we focus on the low-level features and their encoding. We evaluate the use of Fisher vectors as an alternative to bag-of-word histograms to aggregate a small set of state-of-the-art low-level descriptors, in combination with linear classifiers. We present a large and varied set of evaluations, considering (i) classification of short actions in five datasets, (ii) localization of such actions in feature-length movies, and (iii) large-scale recognition of complex events. We find that for basic action recognition and localization MBH features alone are enough for state-of-the-art performance. For complex events we find that SIFT and MFCC features provide complementary cues. On all three problems we obtain state-of-the-art results, while using fewer features and less complex models.*

## 1. Introduction

Action and event recognition in uncontrolled video are extremely challenging due to the large amount of intra-class variation caused by factors such as the style and duration of the performed action. In addition to background clutter and occlusions that are also encountered in image-based recognition, we are confronted with variability due to camera motion, and motion clutter caused by moving background objects. Finally, recognition in video also poses computational challenges due to the sheer amount of data that needs to be processed, particularly so for large-scale datasets.

Recently significant progress has been made in action and event recognition. As a result, the attention of the research community has shifted from relatively controlled settings, as in, *e.g.*, the KTH dataset [37], to more realistic uncontrolled datasets such as the Hollywood2 dataset [23] or the TrecVid Multimedia Event Detection (MED) dataset [29]. At least part of the progress can be attributed to the development of more sophisticated low-level features. Currently, most successful methods are based on some form of local space-time features; see [16, 43] for recent evaluation studies. Most features are carefully engineered, while some recent work explores learning the low-level features from data [20, 45]. Once local features are extracted, often methods similar to those used for object recognition are employed. Typically, local features are quantized, and their overall distribution in a video is represented by means of bag-of-visual-word (BoV) histograms. Possibly, to capture spatio-temporal layout in the spirit of [19], a concatenation of several such histograms is used, which are computed over several space-time cells overlaid on the video [17]. The BoV histograms are then fed into SVM classifiers, often in combination with $\chi^2$-RBF kernels which have been proven to be among the most effective for object recognition.

As for object recognition, the combination of various complementary feature types has been explored. For example, [2] considers feature pooling based on scene-types, where video frames are assigned to scene types and their features are aggregated in the corresponding scene-specific representation. Along similar lines, [9] combines local person and object-centric features, as well as global scene features. Others not only include object detector responses, but also use speech recognition, and character recognition systems to extract additional high-level features [27].

A complementary line of work has focused on considering more sophisticated models for action recognition that go beyond simple bag-of-word representations, and instead aim to explicitly capture the spatial and temporal structure of actions, see *e.g.*, [6, 25]. Other authors have focused on explicitly modeling interactions between people and objects, see *e.g.*, [8, 31], or used multiple instance learning to suppress irrelevant background features [36]. Yet others have used graphical model structures to explicitly model the presence of sub-events [10, 40]. Tang *et al*. [40] use a variable-length discriminative HMM model which infers latent sub-actions together with a non-parametric duration distribution. Izadinia *et al*. [10] use a tree-structured CRF to model co-occurrence relations among sub-events and complex event categories, but require additional labeling of the sub-events unlike Tang *et al*. [40].

Structured models for action recognition seem promising to model basic actions such as *drinking*, *answer phone*, or *get out of car*, which could be decomposed into more basic action units, *e.g*., the "actom" model of Gaidon *et al*. [6]. However, as the definition of the category becomes more high-level, such as *repairing a vehicle tire*, or *making a sandwich*, it becomes less clear to what degree it is possible to learn the structured models from limited amounts of training data, given the much larger amount of within-class variability. Moreover, more complex structured models are generally also computationally more demanding, which limits their usefulness in large-scale settings. To sidestep these potential disadvantages of more complex models, we instead explore the potential of recent advances in robust feature pooling strategies developed in the object recognition literature.

In particular, in this paper we explore the potential of the Fisher vector (FV) encoding [35] as a robust feature pooling technique that has proven to be among the most effective for object recognition [3]. As low-level features we use the dense motion boundary histogram (MBH) features of [41], and evaluate the effect of adding SIFT descriptors to encode appearance information not captured by MBH.

While recently FVs have been explored by others for action recognition [39, 44], we are the first to use them in a large, diverse, and comprehensive evaluation. In parallel to this paper, Jain *et al*. [11] complemented the dense trajectory descriptors with new features computed from optical flow, and encode them using vectors of aggregated local descriptors (VLAD), a simplified version of the Fisher vector. We compare to these works in our experimental evaluation.

We consider three challenging problems. First, we consider the classification of basic action categories using five of the most challenging recent datasets. Second, we consider the localization of actions in feature length movies, using the four actions *drinking*, *smoking*, *sit down*, and *open door* from [4, 18]. Third, we consider classification of more high-level complex event categories using the TrecVid MED 2011 dataset [29]. On all three tasks we obtain state-of-the-art performance, improving over earlier work that relies on combining more feature channels, or using more complex models. For action localization in full length movies we also propose a modified non-maximum-suppression technique that avoids a bias towards selecting shorter segments. This technique further improves the detection performance.

In the next section we present our approach in detail. We present our experimental setup in Section 3, followed by results in Section 4. Finally, we conclude in Section 5.

## 2. Video representation

In this section we first present our feature extraction and encoding pipeline. Then, we discuss how we include weak location information of local features, and finally we discuss non-maximum suppression for action localization.

### 2.1. Feature extraction

We encode the low level visual content using static appearance features as well as motion features. For appearance we use densely extracted SIFT features [22], a method that has been proven extremely successful for object recognition, see *e.g*. [5]. We compute SIFT descriptors every tenth video frame, at multiple scales on a dense grid ($21 \times 21$ patches at 4 pixel steps and 5 scales).

We capture motion information using the recently introduced dense trajectory Motion Boundary Histogram (MBH) features of [41],[1] with default parameters: trajectories of length 15 frames extracted on a dense grid with 5 pixel spacing. The MBH feature is similar to SIFT, but computes gradient orientation histograms over both the vertical and horizontal spatial derivatives of the optical flow. Instead of using a space-time cuboid, MBH descriptors are computed along feature tracks, which ensures that each descriptor is computed from the spatio-temporal volume which follows the motion. Just like in SIFT, gradient orientation histograms are computed in several regular cells along each trajectory, and then concatenated.

### 2.2. Feature encoding

Once the two local low-level features sets are extracted, we use them to construct a signature to characterize the video. For this step we use the Fisher vector (FV) representation [35], which was found to be the most effective one in a recent evaluation study of feature pooling techniques for object recognition [3], which included FVs, bag-of-words, sparse coding techniques, and several variations thereof.

The FV extends the bag-of-visual-words (BoV) representation [38], which is widely used for video classification. The BoV approach is based on the quantization of the local descriptor space using off-line k-means clustering on a large collection of local descriptors. The FV records, for each quantization cell, not only the number of assigned descriptors, but also their mean and variance along each dimension. This leads to a signature with dimension $K(2D + 1)$ for $K$ quantization cells and $D$ dimensional descriptors. Since

[1] We use the implementation publicly available at http://lear.inrialpes.fr/people/wang/dense_trajectories. In parallel to this paper an improved version of the MBH features was developed [42], which corrects the optical flow for camera motion.

more information is stored per cell, a smaller number of quantization cells can be used than for BoV. As the assignment of local descriptors to quantization cells is the main computational cost, the FV signature is faster to compute. Instead of using k-means clustering, Gaussian mixture clustering is used in the FV representation. Local descriptors are then assigned not only to a single quantization cell, but in a weighted manner to multiple clusters using the posterior component probability given the descriptor.

We compute FVs for both SIFT and MBH features. Before computing the FV, we use PCA to project the features to $D = 64$ dimensions. This step speeds up the FV computation and decreases the storage requirements, as the FV size scales linearly with the feature dimension. PCA also decorrelates the data, making the data better fit the diagonal covariance assumption for the Gaussian components. Experiments on the Hollywood2 dataset, using the settings from Table 2, show the performance is stable between 60% to 62% mAP for $D \geq 32$, while it drops to 50% mAP or lower for $D \leq 8$; without PCA the performance is 58.3% mAP. Both the PCA and the GMM are fitted on a subset of $2 \times 10^5$ descriptors from the training dataset.

We apply the power and $\ell_2$ normalizations of [35], which significantly improve the performance in combination with linear classifiers, see results in Table 2. Since the normalization represents a non-linear transformation, it matters when it is applied. For the SIFT features, which are temporally localized in a single frame, we considered two options. First, we compute one FV over the complete video, and then apply the normalization. Second, we compute and normalize a FV per frame, and then average and renormalize the per-frame FVs. In preliminary experiments (results not shown) we found the latter strategy to be more effective, and we use it in all our experiments. For the MBH features we use the first option, since the local features overlap in time.

### 2.3. Weak spatio-temporal location information

To go beyond a completely orderless representation of the video content in a single FV, we consider including a weak notion of spatio-temporal location information of the local features. For this purpose, we use the spatial pyramid (SPM) representation [19], and compute separate FVs over cells in spatio-temporal grids. We also consider the spatial Fisher vector (SFV) of [15], which computes per visual word the mean and variance of the 3D spatio-temporal location of the assigned features. This is similar to extending the (MBH or SIFT) feature vectors with the 3D locations, as done in [26, 34]; the main difference being that the latter do clustering on the extended feature vectors while this is not the case for the SFV. Both methods are complementary, and we combine them by computing SFV in each SPM cell.

The code to aggregate the MBH features in-memory into FVs, and to add SPM and SFV, is available online at `http://lear.inrialpes.fr/software`.

### 2.4. Non-maximum-suppression for localization

For the action localization task we employ a temporal sliding window approach. We score a large pool of candidate detections that are obtained by sliding windows of various lengths across the video. Non-maximum suppression (NMS) is performed to delete windows that have an overlap greater than 20% with higher scoring windows. In practice, we use candidate windows of length $30, 60, 90$, and $120$ frames, and slide the windows in steps of 30 frames.

Preliminary experiments showed that there is a strong tendency for the NMS to retain short windows. This effect is due to the fact that if a relatively long action appears, it is likely that there are short candidate windows that just contain the most characteristic features for the action. Longer windows might better cover the action, but are likely to include less characteristic features (even if they lead to positive classification by themselves), and might include background features due to imperfect temporal alignment.

To address this issue we consider re-scoring the segments by multiplying their score with their duration, before applying NMS (referred to as RS-NMS). We also consider a variant where the goal is to select a subset of candidate windows that (i) covers the video, (ii) does not have overlapping windows, and (iii) maximizes the sum of scores of the selected windows. The optimal subset is found efficiently by dynamic programming as follows. With each time step we associate a state that indicates how long the covering segment is, and where it starts. A pairwise potential is used to enforce consistency: if a segment is not terminated at the current time step, then the next time step should still be covered by the current segment, otherwise a new segment should be started. We use a unary potential that for each state equals the original score of the associated segment. We refer to this method as DP-NMS.

## 3. Experimental setup

Below we present the datasets, evaluation criteria, and the classifier training procedure used in our experiments.

### 3.1. Datasets and evaluation criteria

**Action recognition.** The Hollywood2 [23] dataset is used for a detailed evaluation of the feature encoding parameters. This dataset contains clips of 12 action categories which have been collected from movies. Across all actions there are 810 training samples and 884 test samples; the train and test clips have been selected from different movies. Performance on this data set is measured in terms of mean average precision (mAP) across the categories.

For a comparison to the state of the art we also present experimental results on four of the most challenging action recognition datasets: UCF50 [33], HMDB51 [16], YouTube [21], and Olympics [28]. For these datasets we follow the standard evaluation protocols, as used for example in [41]. We do not repeat them here for the sake of brevity.

**Action localization.** The first dataset we consider for action localization is based on the movie *Coffee and Cigarettes*, and contains annotations for the actions *drinking* and *smoking* [18]. The training set contains 41 and 70 examples from that movie for each class respectively. Additional training examples (32 and 8 respectively) come from the movie *Sea of Love*, and another 33 lab-recorded *drinking* examples are included. The test sets consist of about 20 minutes from *Coffee and Cigarettes* for *drinking*, with 38 positive examples; for *smoking* a sequence of about 18 minutes is used that contains 42 positive examples.

The DLSBP dataset of Duchenne *et al*. [4] contains annotations for the actions *sit down*, and *open door*. The training data comes from 15 movies, and contains 51 *sit down* examples, and 38 for *open door*. The test data contains three full movies (*Living in Oblivion*, *The Crying Game*, and *The Graduate*), which in total last for about 250 minutes, and contain 86 *sit down*, and 91 *open door* samples.

To measure performance we compute the average precision (AP) score as in [4, 6, 13, 18]; considering a detection as correct when it overlaps (as measured by intersection over union) by at least 20% with a ground truth annotation.

**Event recognition.** The TrecVid MED 2011 and 2012 datasets [29] are the largest ones we consider. The 2011 dataset consists of consumer videos from 15 categories that are more complex than the basic actions considered in the other datasets, *e.g*. *changing a vehicle tire*, or *birthday party*. For each category between 100 and 300 training videos are available. In addition, 9,600 videos are available that do not contain any of the 15 categories; this data is referred to as the *null* class. The test set consists of 32,000 videos, with a total length of about 1,000 hours, and includes 30,500 videos of the *null* class.

We follow two experimental setups in order to compare our system to previous work. The first setup is the one described above, which was also used in the TrecVid 2011 MED challenge; performance is evaluated using the minimum Normalized Detection Costs (min-NDC) measure. The NDC is a weighted linear combination of the missed detection and false alarm probabilities, and the minimum is taken over possible decision thresholds, see [29]. We also report results with the standard mean average precision (AP) measure. The second setup is the one of Sun *et al*. [39]. Their split contains 13,274 videos: 8,840 for training and 4,434 for testing. These videos were randomly selected from the MED 2011 and 2012 data. Thus, there are 25 categories for this setup, corresponding to the number of categories in MED'12. The list of videos used for training and testing was obtained through personal communication with the authors of [39].

The videos in the TrecVid dataset vary strongly in size: durations range from a few seconds to one hour, while the resolution ranges from low quality $128 \times 88$ to full HD $1920 \times 1080$. We rescale the videos to a width of at most 200 pixels preserving the aspect ratio and temporally sub-sample them by discarding every second frame. These rescaling parameters were selected on a subset of the MED data: compared to other less severe rescaling (*e.g*., width at most 800 pixels and no temporal sub-sampling), we get similar performance, while speeding-up by more than an order of magnitude. The number of extracted features is roughly proportional to the video size; therefore video rescaling linearly speeds up the feature extraction and encoding time. Our complete pipeline —video re-scaling, feature extraction and feature encoding— runs at 2.4 slower than real-time on a single core.

### 3.2. Classifier training

In all experiments we train linear SVM classifiers, and set the regularization parameter by cross-validation. We weight positive and negative examples inversely proportional to the number of corresponding samples, so that both classes effectively contain the same number of examples.

When using BoV histograms we use the part of the FV that corresponds to the derivatives of the mixing weights, and still apply power and $\ell_2$ normalizations in combination with linear classifiers. The power normalization can be seen as an approximate explicit embedding of the $\chi^2$ kernel [30].

When using multiple features, we employ a late fusion strategy and linearly combine classifier scores computed for each feature. We perform a grid-search over the weights, and use cross-validation to directly optimize with respect to the relevant evaluation metric.

## 4. Experimental results

In our experimental evaluation below, we consider the three different problems described above in turn.

### 4.1. Action recognition experiments

In our first set of experiments we only use the MBH descriptor and compare the Fisher Vector (FV) and Bag-of-visual-word (BoV) encoding, for dictionaries from 50 up to 4000 visual words. We also evaluate the effect of including weak geometric information using the spatial Fisher vector (SFV) and spatio-temporal grids (SPM). We consider SPM grids that divide the video in two temporal parts (T2), and/or spatially in three horizontal bins (H3). When using SPM we always concatenate the representations with the FVs computed over the whole video, so when we use T2+H3 we concatenate six FVs in total (one for the whole image, two for T2, and three for H3). Note that for FVs

|  |  | Bag-of-words |  | Fisher vectors |  |
|---|---|---|---|---|---|
| $K$ | SPM | — | SFV | — | SFV |
| 50 | — | 38.7 | 43.4 | 52.1 | 54.2 |
| 50 | H3 | 38.6 | 44.5 | 55.2 | 56.9 |
| 50 | T2 | 43.2 | 45.7 | 56.2 | 57.2 |
| 50 | T2+H3 | 43.9 | 46.8 | 57.7 | 58.8 |
| 100 | — | 41.5 | 45.1 | 55.9 | 57.5 |
| 100 | T2 | 43.1 | 48.1 | 57.4 | 58.7 |
| 100 | H3 | 45.5 | 47.1 | 57.9 | 59.1 |
| 100 | T2+H3 | 47.0 | 50.0 | 59.1 | 60.1 |
| 500 | — | 46.1 | 51.6 | 57.7 | 59.0 |
| 500 | H3 | 47.9 | 53.3 | 58.8 | 60.2 |
| 500 | T2 | 47.8 | 53.1 | 59.6 | 60.1 |
| 500 | T2+H3 | 50.7 | 53.8 | 60.5 | 61.5 |
| 1000 | — | 47.7 | 53.3 | 58.2 | 59.8 |
| 1000 | H3 | 49.7 | 55.3 | 59.2 | 60.5 |
| 1000 | T2 | 49.5 | 54.8 | 60.0 | 60.9 |
| 1000 | T2+H3 | 52.4 | 56.1 | 60.7 | 61.9 |
| 4000 | — | 51.3 | 56.2 | 57.5 | 59.2 |
| 4000 | H3 | 54.5 | 57.7 | 57.0 | 58.8 |
| 4000 | T2 | 55.1 | 57.7 | 59.1 | 60.0 |
| 4000 | T2+H3 | 56.5 | 58.1 | 59.2 | 60.0 |

Table 1. Comparison of FV and BoV on the Hollywood2 dataset using MBH features only, and varying the number of Gaussians ($K$), and using SPM and SFV to include location information.

the SFV has only a limited effect on the representation size, as it just adds six dimensions (for the spatio-temporal means and variances) for each visual word, on top of the $64 + 64 + 1 = 129$ dimensional gradient vector computed for the mixing weights, means and variances in the descriptor space. For the BoV representation the situation is quite different, since in that case there is only a single count per visual word, and the additional six dimensions of the SFV multiply the signature size by a factor of seven.

In Table 1 we present the performance of the different settings in terms of the mAP. Generally across all settings, performance is increasing with the number of Gaussians, and FVs lead to significantly better performance than BoV. Both BoV and FV benefit from including SPM and SFV, which are complementary since best performance is always obtained when they are combined. SFV is relatively more effective for BoV than for FV, probably because it has a larger impact on the signature dimensionality for the former.

Our experiments show that FVs using 50 visual words are comparable to BoV histograms for 4000 visual words; confirming that for FVs fewer visual words are needed than for BoV histograms. This shows that FVs are more efficient for large-scale applications, since the feature encoding step is one of the main computational bottlenecks and it scales linearly with the dictionary size.

Using the best setting from these experiments, FVs with SFV+T2+H3 and $K = 1000$, we now compare our results to the state of the art in Table 2 on five action recognition datasets. On all datasets our performance is comparable or better than the current state of the art using only MBH features. The SIFT features perform significantly worse, and carry relatively little useful complementary information.

|  |  |  | UCF50 | HMBD51 | YouTube | Olympics | Hollywood2 |
|---|---|---|---|---|---|---|---|
|  | $\sqrt{\cdot}$ | $\ell_2$ |  |  |  |  |  |
| MBH | N | N | 83.8 | 42.6 | 84.4 | 77.7 | 53.7 |
|  | N | Y | 86.1 | 46.5 | 86.8 | 78.4 | 58.9 |
|  | Y | N | 86.2 | 45.5 | 86.2 | 80.9 | 61.5 |
| MBH | Y | Y | 87.8 | 51.9 | 88.5 | **84.6** | 61.9 |
| SIFT | Y | Y | 76.3 | 34.8 | 77.2 | 58.7 | 42.5 |
| MBH+SIFT | Y | Y | **90.0** | **54.8** | **89.0** | 82.1 | **63.3** |
| BT'10 |  | [1] | — | — | 77.8 | — | — |
| LZYN'11 |  | [20] | — | — | 75.8 | — | 53.3 |
| KGGHW'12 |  | [14] | 72.7 | 29.2 | — | — | — |
| WWQ'12 |  | [44] | — | 31.8 | — | — | — |
| JDXLN'12 |  | [12] | — | 40.7 | — | 80.6 | 59.5 |
| GHS'12 |  | [7] | — | — | — | 82.7 | — |
| MS'12 |  | [24] | — | — | — | — | 61.7 |
| WKSCL'13 |  | [41] | 85.6 | 48.3 | 85.4 | 77.2 | 59.9 |
| JJB'13 |  | [11] | — | 52.1 | — | 83.2 | 62.5 |

Table 2. Comparison to the state of the art of our FV-based results with SFV+T2+H3, $K = 1000$ for both MBH and SIFT features. For the MBH features we show the impact of the signed square root normalization ($\sqrt{\cdot}$) and $\ell_2$ normalization ($\ell_2$).

The comparison to [41] shows the effectiveness of the FV representation: they used 4000 visual words with $\chi^2$-RBF kernels and in addition to MBH also included HOG, HOF and trajectory features as well as a spatio-temporal grid. Le *et al*. [20], learn spatio-temporal features directly using a convolutional network, instead of relying on designed features. Brendel *et al*. [1] represent videos as a temporal sequence of poses and use an exemplar-based recognition at test time. Kliper-Gros *et al*. [14] encode local motion patterns by matching patches across successive video frames, and aggregate the quatized motion patterns in a BoV reprensentation. Wang *et al*. [44] uses sparse coding with sum-pooling over the STIP+HOG/HOF features of [17], which they found to work slightly better than FVs (albeit using 64 times fewer visual words for the FVs). Jiang *et al*. [12] use the dense trajectory features of [41] and use an extended BoV encoding over pairs of local features to explicitly cancel common (camera) motion patterns. The results of Gaidon *et al*. [7] are based on a hierarchical clustering of dense trajectories of [41] and concatenated BoV representations over child and parent nodes in the clustering hierarchy. Mathe and Sminchisescu [24] use multiple-kernel learning to combine 14 descriptors sampled on human attention maps with the dense trajectory features of [41]. Jain

|  | Overlap | Drinking | Smoking | Open door | Sit Down |
|---|---|---|---|---|---|
| NMS | 20 | 56.5 | 42.8 | **27.0** | 17.0 |
| RS-NMS | 20 | 61.9 | 48.7 | 23.6 | 17.7 |
| DP-NMS | 0 | 53.9 | 47.1 | 23.7 | 14.1 |
| NMS-0 | 0 | 54.4 | 42.9 | 26.9 | 17.2 |
| RS-NMS-0 | 0 | **63.9** | **50.5** | 26.5 | **18.2** |

Table 3. Evaluation of the NMS variants for action localization.

*et al*. [11] use camera motion stabilization, and use VLADs to aggregate local MBH, HOG, HOF, and their novel kinematic Divergence-Curl-Shear flow features.

### 4.2. Action localization experiments

In our second set of experiments we consider the localization of four actions in feature length movies. Given the size of the test dataset, we encode both the MBH and SIFT features with FVs with $K = 128$ Gaussians and do not include location information with SPM or SFV.

First, we consider the effect of the different NMS variants in Table 3 using MBH features alone. We see that simple rescoring (RS-NMS) compares favorably to standard NMS on three out of four classes, while the dynamic-programming version (DP-NMS) improves on a single class, but deteriorates on three. To test whether this is due to the fact that DP-NMS does not allow any overlap, we also test NMS and RS-NMS with zero-overlap. The results show that for standard NMS zero or 20% overlap does not significantly change the results on three out of four classes, while for RS-NMS zero overlap is beneficial on all classes.

In Table 4 we compare our results for the RS-NMS-0 method with previously reported results. On three of the four actions we obtain substantially better results, despite the fact that previous work used more elaborate techniques. For example, [13] relied on a person detector, while [6] requires finer annotations that indicate the position of characteristic moments of the actions (actoms).

As for the action recognition datasets, we also find that the SIFT features carry little complementary information, and are actually detrimental when combined with the MBH features by late fusion. The negative impact on performance might be due to the small training datasets used here, which might render the late-fusion process unstable.

### 4.3. Event recognition experiments

In our last set of experiments we consider the TrecVid MED 2011 event recognition dataset. In Table 5 we provide a detailed per-event evaluation of the MBH and SIFT features, as well as their combination. For both features we use $K = 256$ visual words, and exclude SPM and SFV for efficiency. In this case the SIFT and MBH features are highly complementary as their combination leads to significant performance improvements.

|  | Drinking | Smoking | Open door | Sit Down |
|---|---|---|---|---|
| LP'07 [18] | 49 | — | — | — |
| DLSBP'09 [4] | 40 | — | 14.4 | 13.9 |
| KMSZ'10 [13] | 54.1 | 24.5 | — | — |
| GHS'11 [6] | 57 | 31 | 16.4 | **19.8** |
| MBH | **63.9** | **50.5** | **26.5** | 18.2 |
| SIFT | 22.1 | 20.7 | 10.6 | 11.0 |
| MBH+SIFT | 56.6 | 43.0 | 23.2 | 16.7 |

Table 4. Action localization performance with RS-NMS-0 and different features compared to earlier work.

We compare our results to the best submitted run[2] in the 2011 MED challenge [27], which outperforms our SIFT+MBH results on six of the ten classes using the min-NDC measure. It should be noted, however, that Natarajan *et al*. [27] combine many features from different modalities, including audio features, and high-level features obtained from object detector responses, automatic speech recognition, and video text recognition.

We did not want to include any high-level features, since that implies employing external training data, which renders any comparison more difficult. We did, however, experiment with adding audio features: the mel-frequency cepstral coefficients (MFCC) and their first and second derivatives [32]. The concatenation of these three parts, each having 13 dimensions, yielded a 39-dimensional vector. We follow exactly the same FV encoding scheme as used before for the MBH and SIFT features, using $K = 512$. With the inclusion of the audio features our results are comparable or better on eight of the ten categories, and also better on average. For completeness and better readability, we include the AP scores for our results in the same table.

Finally, we compare to the results reported in [39] in Table 6, using the second evaluation setup described in Section 3.1. Our results significantly outperform theirs by 8% mAP without using the MFCC audio features. The results obtained with the MBH features are comparable to theirs. Sun *et al*. [39] also use FVs for dense trajectories, but include four types of descriptors (MBH, HOG, HOF and the shape of the trajectories) as well as use a spatial pyramid and a Gaussian kernel, whereas we only use FVs with MBH descriptors and linear classifiers, but use more visual words. This is significantly faster, which is important if the entire MED dataset is used and not only a subset. Our results follow a similar trend as in the previous experiment, Table 5: the main gain is due to the SIFT descriptors (8% mAP) and adding the audio further increases the score by 4% mAP.

[2] This run is referred to as `BBNVISER_c-Fusion2_2` in MED 2011.

|  |  | Birthday party | Changing a vehicle tire | Flash mob gathering | Getting vehicle unstuck | Grooming an animal | Making a sandwich | Parade | Parkour | Repairing an appliance | Sewing project | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| min-NDC | Best MED 2011 entry [27] | **0.446** | 0.475 | 0.280 | 0.379 | 0.622 | **0.570** | 0.446 | 0.308 | 0.381 | 0.575 | 0.448 |
|  | MBH | 0.766 | 0.785 | 0.338 | 0.590 | 0.754 | 0.768 | 0.523 | 0.254 | 0.531 | 0.652 | 0.596 |
|  | SIFT | 0.713 | 0.627 | 0.400 | 0.452 | 0.746 | 0.693 | 0.713 | 0.570 | 0.611 | 0.768 | 0.629 |
|  | MBH+SIFT | 0.624 | 0.543 | **0.256** | **0.369** | 0.666 | 0.618 | 0.445 | 0.223 | 0.461 | 0.604 | 0.481 |
|  | MBH+SIFT+MFCC | 0.488 | **0.480** | 0.261 | 0.377 | **0.586** | 0.646 | **0.414** | **0.214** | **0.351** | **0.517** | **0.434** |
| AP | MBH | 20.40 | 15.50 | 54.76 | 30.43 | 18.33 | 13.10 | 41.21 | 71.03 | 34.56 | 29.15 | 32.84 |
|  | SIFT | 23.10 | 28.88 | 48.49 | 31.76 | 17.12 | 17.09 | 30.27 | 37.50 | 33.20 | 22.95 | 29.04 |
|  | MBH+SIFT | 27.37 | 34.59 | 61.94 | **41.64** | 21.37 | 20.21 | 47.88 | 71.43 | 43.55 | 34.57 | 40.45 |
|  | MBH+SIFT+MFCC | **45.33** | **41.77** | **63.90** | 39.16 | **27.24** | **21.64** | **53.22** | **71.91** | **50.85** | **38.20** | **45.32** |

Table 5. Performance in terms of min-NDC and AP on the TrecVid MED 2011 dataset, and comparison to the best entry in MED 2011. Note that for min-NDC lower is better.

|  | SN'13 [39] | MBH | SIFT | MBH+SIFT | MBH+SIFT +MFCC |
|---|---|---|---|---|---|
| Attempting board trick | 50.7 | 50.7 | 44.8 | 55.5 | **59.5** |
| Feeding an animal | 15.9 | 12.9 | 11.9 | 20.9 | **22.1** |
| Landing a fish | 44.7 | 32.4 | 43.5 | 45.9 | **49.3** |
| Wedding ceremony | 61.0 | 57.7 | 70.8 | **72.9** | 72.2 |
| Woodworking project | 29.3 | 18.3 | 30.7 | 41.6 | **52.3** |
| Birthday party | 30.9 | 20.0 | 28.2 | 31.2 | **43.8** |
| Changing a vehicle tire | 28.0 | 26.2 | 30.1 | 37.9 | **38.9** |
| Flash mob gathering | 57.6 | 57.6 | 53.5 | 61.9 | **64.2** |
| Getting vehicle unstuck | 46.9 | 45.4 | 51.4 | 57.9 | **60.0** |
| Grooming an animal | 29.5 | 20.5 | 32.8 | **37.6** | 36.9 |
| Making a sandwich | 25.6 | 36.8 | 32.3 | 43.2 | **44.6** |
| Parade | 51.7 | 48.5 | 35.7 | **54.9** | 53.9 |
| Parkour | 48.3 | 60.1 | 46.2 | 65.7 | **66.1** |
| Repairing an appliance | 45.7 | 46.8 | 47.9 | 56.0 | **65.1** |
| Sewing project | 47.1 | 49.9 | 33.3 | 56.1 | **62.6** |
| Attempting a bike trick | 49.1 | 46.4 | 43.9 | 60.0 | **63.6** |
| Cleaning an appliance | 9.1 | 11.2 | 11.3 | 15.9 | **25.5** |
| Dog show | 67.4 | 75.6 | 60.6 | **77.1** | 75.8 |
| Giving directions to a location | 9.3 | 17.1 | 13.0 | 13.5 | **27.0** |
| Marriage proposal | 14.0 | 23.6 | 4.9 | 24.1 | **31.7** |
| Renovating a home | 44.7 | 26.5 | 39.4 | **41.6** | 36.2 |
| Rock climbing | 56.6 | 43.7 | 38.8 | 54.5 | **56.8** |
| Town hall meeting | 45.0 | 48.0 | 44.6 | 55.1 | **78.8** |
| Winning a race without a vehicle | 27.5 | 35.2 | 26.4 | **36.9** | 33.6 |
| Working on a metal crafts project | 10.1 | 12.9 | 18.3 | 22.7 | **22.9** |
| mAP | 37.8 | 37.0 | 35.8 | 45.6 | **49.7** |

Table 6. Evaluation on TrecVid MED using the protocol of [39].

## 5. Conclusions

We presented an efficient action recognition system that combines three state-of-the-art low-level descriptors (MBH, SIFT, MFCC) with the recent Fisher vector representation. In our experimental evaluation we considered action recognition, action localization in movies, and complex event recognition. For the first two tasks, we observed that MBH motion features carry much more discriminative information than SIFT features, and that the latter bring little or no complementary information. A detailed evaluation on the Hollywood2 action recognition dataset showed the effectiveness and complementarity of SPM and SFV to include weak geometric information, and that FVs provide a more efficient feature encoding method than BoV histograms since fewer visual words are needed. We found that action localization results can be substantially improved by using a simple re-scoring technique before applying NMS, to suppress a bias for shorter windows. For recognition of event categories, we find that the SIFT features do bring useful contextual information, as do MFCC audio features.

Our experimental evaluation is among the most extensive and diverse ones to date, including five of the most challenging action recognition benchmarks, action localization in feature length movies, and large-scale event recognition with a test set of more than 1,000 hours of video. Across all these datasets the combination of FVs with state-of-the-art descriptors outperforms the current state of the art, while using less features and less complex models. Therefore we believe that, currently, the presented system is the most effective one for deployment in large-scale action and event recognition problems, such as encountered in practice in broadcast archives or user-generated content archives.

**Acknowledgements.** This work was supported by Quaero (funded by OSEO, French State agency for innovation), the European integrated project AXES, the MSR/INRIA joint project and the ERC advanced grant ALLEGRO.

## References

[1] W. Brendel and S. Todorovic. Activities as time series of human postures. In *ECCV*, 2010.

[2] L. Cao, Y. Mu, A. Natsev, S.-F. Chang, G. Hua, and J. Smith. Scene aligned pooling for complex video recognition. In

*ECCV*, 2012.

[3] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011.

[4] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *ICCV*, 2009.

[5] M. Everingham, L. van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, June 2010.

[6] A. Gaidon, Z. Harchaoui, and C. Schmid. Actom sequence models for efficient action detection. In *CVPR*, 2011.

[7] A. Gaidon, Z. Harchaoui, and C. Schmid. Recognizing activities with cluster-trees of tracklets. In *BMVC*, 2012.

[8] A. Gupta, A. Kembhavi, and L. Davis. Observing human-object interactions: using spatial and functional compatibility for recognition. *PAMI*, 31(10):1775–1789, 2009.

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

[10] H. Izadinia and M. Shah. Recognizing complex events using large margin joint low-level event model. In *ECCV*, 2012.

[11] M. Jain, H. Jégou, and P. Bouthemy. Better exploiting motion for better action recognition. In *CVPR*, 2013.

[12] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo. Trajectory-based modeling of human actions with motion reference points. In *ECCV*, 2012.

[13] A. Kläser, M. Marszałek, C. Schmid, and A. Zisserman. Human focused action localization in video. In *ECCV Workshop on Sign, Gesture, and Activity*, 2010.

[14] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *ECCV*, 2012.

[15] J. Krapac, J. Verbeek, and F. Jurie. Modeling spatial layout with Fisher vectors for image categorization. In *ICCV*, 2011.

[16] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

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

[18] I. Laptev and P. Pérez. Retrieving actions in movies. In *ICCV*, 2007.

[19] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

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

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

[22] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[23] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009.

[24] S. Mathe and C. Sminchisescu. Dynamic eye movement datasets and learnt saliency models for visual action recognition. In *ECCV*, 2012.

[25] P. Matikainen, M. Hebert, and R. Sukthankar. Representing pairwise spatial and temporal relations for action recognition. In *ECCV*, 2010.

[26] S. McCann and D. Lowe. Spatially local coding for object recognition. In *ACCV*, 2012.

[27] P. Natarajan, S. Wu, S. Vitaladevuni, X. Zhuang, S. Tsakalidis, U. Park, R. Prasad, and P. Natarajan. Multimodal feature fusion for robust event detection in web videos. In *CVPR*, 2012.

[28] J. Niebles, C.-W. Chen, and F.-F. Li. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010.

[29] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, B. Shaw, W. Kraaij, A. F. Smeaton, and G. Quenot. TRECVID 2012 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID*, 2012.

[30] F. Perronnin, J. Sánchez, and Y. Liu. Large-scale image categorization with explicit data embedding. In *CVPR*, 2010.

[31] A. Prest, V. Ferrari, and C. Schmid. Explicit modeling of human-object interactions in realistic videos. *PAMI*, 35(4):835–848, 2013.

[32] L. Rabiner and R. Schafer. Introduction to digital speech processing. *Foundations and Trends in Signal Processing*, 1(1/2):1–194, 2007.

[33] K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. In *Machine Vision and Applications Journal*, 2012.

[34] J. Sánchez, F. Perronnin, and T. de Campos. Modeling the spatial layout of images beyond spatial pyramids. *Pattern Recognition Letters*, 33(16):2216–2223, 2012.

[35] J. Sanchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the Fisher vector: Theory and practice. *IJCV*, 105(3):222–245, June 2013.

[36] M. Sapienza, F. Cuzzolin, and P. Torr. Learning discriminative space-time actions from weakly labelled videos. In *BMVC*, 2012.

[37] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *ICPR*, 2004.

[38] J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. In *ICCV*, 2003.

[39] C. Sun and R. Nevatia. Large-scale web video event classification by use of Fisher vectors. In *WACV*, 2013.

[40] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *CVPR*, 2012.

[41] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 2013.

[42] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

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

[44] X. Wang, L. Wang, and Y. Qiao. A comparative study of encoding, pooling and normalization methods for action recognition. In *ACCV*, 2012.

[45] Y. Yang and M. Shah. Complex events detection using data-driven concepts. In *ECCV*, 2012.