Int J Comput Vis (2016) 119:219–238
DOI 10.1007/s11263-015-0846-5

A Robust and Efficient Video Representation for Action Recognition

Heng Wang[1] · **Dan Oneata**[2] · **Jakob Verbeek**[2] · **Cordelia Schmid**[2]

Received: 15 June 2014 / Accepted: 4 July 2015 / Published online: 17 July 2015

Abstract This paper introduces a state-of-the-art video representation and applies it to efficient action recognition and detection. We first propose to improve the popular dense trajectory features by explicit camera motion estimation. More specifically, we extract feature point matches between frames using SURF descriptors and dense optical flow. The matches are used to estimate a homography with RANSAC. To improve the robustness of homography estimation, a human detector is employed to remove outlier matches from the human body as human motion is not constrained by the camera. Trajectories consistent with the homography are considered as due to camera motion, and thus removed. We also use the homography to cancel out camera motion from the optical flow. This results in significant improvement on motion-based HOF and MBH descriptors. We further explore the recent Fisher vector as an alternative feature encoding approach to the standard bag-of-words (BOW) histogram, and consider different ways to include spatial layout information in these encodings. We present a large and varied set of evaluations, considering (i) classification of short basic actions on six datasets, (ii) localization of such actions in feature-length movies, and (iii) large-scale recognition of complex events. We find that our improved trajectory features significantly outperform previous dense trajectories, and that Fisher vectors are superior to BOW encodings for video recognition tasks. In all three tasks, we show substantial improvements over the state-of-the-art results.

Keywords Action recognition · Action detection · Multimedia event detection

Communicated by Ivan Laptev, Josef Sivic, and Deva Ramanan.

H. Wang is currently with Amazon Research Seattle, but carried out the work described here while being affiliated with INRIA.

✉ Dan Oneata
dan.oneata@inria.fr

Heng Wang
hengwang00@gmail.com

Jakob Verbeek
Jakob.Verbeek@inria.fr

Cordelia Schmid
Cordelia.Schmid@inria.fr

[1] Amazon Research, Seattle, WA, USA

[2] INRIA, Grenoble, France

1 Introduction

Action and event recognition have been an active research topic for over three decades due to their wide applications in video surveillance, human computer interaction, video retrieval, etc. Research in this area used to focus on simple datasets collected from controlled experimental settings, eg, the KTH (Schüldt et al. 2004) and Weizmann (Gorelick et al. 2007) datasets. Due to the increasing amount of video data available from both internet repositories and personal collections, there is a strong demand for understanding the content of real world complex video data. As a result, the attention of the research community has shifted to more realistic datasets such as the Hollywood2 dataset (Marszałek et al. 2009) or the TRECVID multimedia event detection (MED) dataset (Over et al. 2012).

The diversity of realistic video data has resulted in different challenges for action and event recognition. First, there is tremendous 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

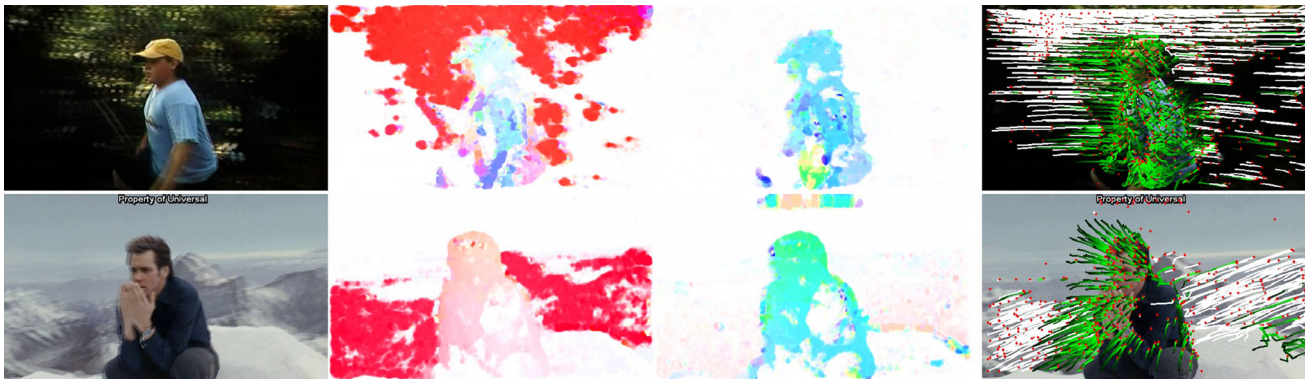

Fig. 1 *First column* images of two consecutive frames overlaid; *second column* optical flow (Farnebäck 2003) between the two frames; *third column* optical flow after removing camera motion; *last column* trajectories removed due to camera motion in white

variability due to camera motion, and motion clutter caused by moving background objects. Challenges can also come from the low quality of video data, such as noise due to the sensor, camera jitter, various video decoding artifacts, etc. 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 such as the 2014 edition of the TRECVID MED dataset which contains over 8000 h of video.

Local space-time features (Dollár et al. 2005; Laptev 2005) have been shown to be advantageous in handling such datasets, as they allow to directly build efficient video representations without non-trivial pre-processing steps, such as object tracking or motion segmentation. 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 with bag-of-words (BOW) histograms, see, eg, (Kuehne et al. 2011; Wang et al. 2009) for recent evaluation studies.

The success of local space-time features leads to a trend of generalizing classical descriptors from image to video, eg, 3D-SIFT (Scovanner et al. 2007), extended SURF (Willems et al. 2008), HOG3D (Kläser et al. 2008), and local trinary patterns (Yeffet and Wolf 2009). Among the local space-time features, dense trajectories (Wang et al. 2013a) have been shown to perform the best on a variety of datasets. The main idea is to densely sample feature points in each frame, and track them in the video based on optical flow. Multiple descriptors are computed along the trajectories of feature points to capture shape, appearance and motion information. Interestingly, motion boundary histograms (MBH) (Dalal et al. 2006) give the best results due to their robustness to camera motion.

MBH is based on derivatives of optical flow, which is a simple and efficient way to achieve robustness to camera motion. However, MBH only suppresses certain camera motions and, thus, we can benefit from explicit camera motion estimation. Camera motion generates many irrelevant trajectories in the background in realistic videos. We can prune them and only keep trajectories from humans and objects of interest, if we know the camera motion, see Fig. 1. Furthermore, given the camera motion, we can correct the optical flow, so that the motion vectors from human body are independent of camera motion. This improves the performance of motion descriptors based on optical flow, i.e., histograms of optical flow (HOF) and MBH. We illustrate the difference between the original and corrected optical flow in the middle two columns of Fig. 1.

Besides improving low-level video descriptors, we also employ Fisher vectors (Sánchez et al. 2013) to encode local descriptors into a holistic representation. Fisher vectors have been shown to give superior performance over BOW in image classification (Chatfield et al. 2011; Sánchez et al. 2013). Our experimental results prove that the same conclusion also holds for a variety of recognition tasks in the video domain.

We consider three challenging problems to demonstrate the effectiveness of our proposed framework. First, we consider the classification of basic action categories using six of the most challenging datasets. Second, we consider the localization of actions in feature length movies, including four action classes: *drinking, smoking, sit down*, and *open door* from (Duchenne et al. 2009; Laptev and Pérez 2007). Third, we consider classification of more high-level complex event categories using the TRECVID MED 2011 dataset (Over et al. 2012).

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 short segments, and further improves the detection performance. This paper integrates and extends our previous results which have appeared in earlier papers

(Oneata et al. 2013; Wang and Schmid 2013). The code to compute improved trajectories and descriptors is available online.[1]

The rest of the paper is organized as follows. Section 2 reviews related work. We detail our improved trajectory features by explicit camera motion estimation in Sect. 3. Feature encoding and non-maximum-suppression for action localization are presented in Sects. 4 and 5. Datasets and evaluation protocols are described in Sect. 6. Experimental results are given in Sect. 7. Finally, we present our conclusions in Sect. 8.

2 Related Work

Feature trajectories (Matikainen et al. 2009; Messing et al. 2009; Sun et al. 2009; Wang et al. 2013a) have been shown to be a good way for capturing the intrinsic dynamics of video data. Very few approaches consider camera motion when extracting feature trajectories for action recognition. Uemura et al. (2008) combine feature matching with image segmentation to estimate the dominant camera motion, and then separate feature tracks from the background. Wu et al. (2011) apply a low-rank assumption to decompose feature trajectories into camera-induced and object-induced components. Gaidon et al. (2013) use efficient image-stitching techniques to compute the approximate motion of the background plane and generate stabilized videos before extracting dense trajectories (Wang et al. 2013a) for activity recognition.

Camera motion has also been considered in other types of video representations. Ikizler-Cinbis and Sclaroff (2010) use of a homography-based motion compensation approach in order to estimate the foreground optical flow field. Li et al. (2012) recognize different camera motion types such as pan, zoom and tilt to separate foreground and background motion for video retrieval and summarization. Recently, Park et al. (2013) perform weak stabilization to remove both camera and object-centric motion using coarse-scale optical flow for pedestrian detection and pose estimation in video.

Due to the excellent performance of dense trajectories on a wide range of action datasets (Wang et al. 2013a), there are several approaches try to improve them from different perspectives. Vig et al. (2012) propose to use saliency-mapping algorithms to prune background features. This results in a more compact video representation, and improves action recognition accuracy. Jiang et al. (2012) cluster dense trajectories, and use the cluster centers as reference points so that the relationship between them can be modeled. Jain et al. (2013) decompose visual motion into dominant and residual motions both for extracting trajectories and computing descriptors.

Besides carefully engineering video features, some recent work explores learning low-level features from video data (Le et al. 2011; Yang and Shah 2012). For example, Cao et al. (2012) consider 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, Ikizler-Cinbis and Sclaroff (2010) 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 (Natarajan et al. 2012).

A complementary line of work has focused on considering more sophisticated models for action recognition that go beyond simple BOW representations, and aimed to explicitly capture the spatial and temporal structure of actions, see eg, (Gaidon et al. 2011; Matikainen et al. 2010). Other authors have focused on explicitly modeling interactions between people and objects, see eg, (Gupta et al. 2009; Prest et al. 2013), or used multiple instance learning to suppress irrelevant background features (Sapienza et al. 2012). Yet others have used graphical model structures to explicitly model the presence of sub-events (Izadinia and Shah 2012; Tang et al. 2012). Tang et al. (2012) use a variable-length discriminative HMM model which infers latent sub-actions together with a non-parametric duration distribution. Izadinia and Shah (2012) 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. (2012).

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, eg, the "actom" model of Gaidon et al. (2011). 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 intra-class variability. Moreover, more complex structured models are generally more computationally 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 encoding (Sánchez et al. 2013) as a robust feature pooling technique that has been proven to be among the most effective for object recognition (Chatfield et al. 2011). While recently Fisher vectors (FVs) have been explored by

[1] http://lear.inrialpes.fr/~wang/improved_trajectories.

others for action recognition (Sun and Nevatia 2013; Wang et al. 2012), we are the first to use them in a large, diverse, and comprehensive evaluation. In parallel to this paper, Jain et al. (2013) complemented the dense trajectory descriptors with new features computed from optical flow, and encoded them using vectors of locally aggregated descriptors (VLAD; Jégou et al. 2011), a simplified version of the Fisher vector. We compare to these works in our experimental evaluation.

3 Improving Dense Trajectories

In this section, we first briefly review the dense trajectory features (Wang et al. 2013a). We, then, detail the major steps of our improved trajectory features including camera motion estimation, removing inconsistent matches using human detection, and extracting improved trajectory features, respectively.

3.1 Dense Trajectory Features

The dense trajectory features approach (Wang et al. 2013a) densely samples feature points for several spatial scales. Points in homogeneous areas are suppressed, as it is impossible to track them reliably. Tracking points is achieved by median filtering in a dense optical flow field (Farnebäck 2003). In order to avoid drifting, we only track the feature points for 15 frames and sample new points to replace them. We remove static feature trajectories as they do not contain motion information, and also prune trajectories with sudden large displacements.

For each trajectory, we compute HOG, HOF and MBH descriptors with exactly the same parameters as in Wang et al. (2013a). Note that we do not use the trajectory descriptor as it does not improve the overall performance significantly. All three descriptors are computed in the space-time volume aligned with the trajectory. HOG (Dalal and Triggs 2005) is based on the orientation of image gradients and captures the static appearance information. Both HOF (Laptev et al. 2008) and MBH (Dalal et al. 2006) measure motion information, and are based on optical flow. HOF directly quantizes the orientation of flow vectors. MBH splits the optical flow into horizontal and vertical components, and quantizes the derivatives of each component. The final dimensions of the descriptors are 96 for HOG, 108 for HOF and 2×96 for the two MBH channels.

To normalize the histogram-based descriptors, i.e., HOG, HOF and MBH, we apply the recent RootSIFT (Arandjelovic and Zisserman 2012) approach, i.e., square root each dimension after ℓ_1 normalization. We do not perform ℓ_2 normalization as in Wang et al. (2013a). This slightly improves the results without introducing additional computational cost.

3.2 Camera Motion Estimation

To estimate the global background motion, we assume that two consecutive frames are related by a homography (Szeliski 2006). This assumption holds in most cases as the global motion between two frames is usually small. It excludes independently moving objects, such as humans and vehicles.

To estimate the homography, the first step is to find the correspondences between two frames. We combine two approaches in order to generate sufficient and complementary candidate matches. We extract speeded-up robust features (SURF; Bay et al. 2006) and match them based on the nearest neighbor rule. SURF features are obtained by first detecting interest points based on an approximation of the Hessian matrix and then describing them by a distribution of Haar-wavelet responses. The reason for choosing SURF features is their robustness to motion blur, as shown in a recent evaluation (Gauglitz et al. 2011).

We also sample motion vectors from the optical flow, which provides us with dense matches between frames. Here, we use an efficient optical flow algorithm based on polynomial expansion (Farnebäck 2003). We select motion vectors for salient feature points using the good-features-to-track criterion (Shi and Tomasi 1994), i.e., thresholding the smallest eigenvalue of the autocorrelation matrix. Salient feature points are usually reproducible (stable under local and global perturbations, such as illumination variations or geometric transformation) and distinctive (with rich local structure information). Motion estimation on salient points is more reliable.

The two approaches are complementary. SURF focuses on blob-type structures, whereas (Shi and Tomasi 1994) fires on corners and edges. Figure 2 visualizes the two types of matches in different colors. Combining them results in a more

Fig. 2 Visualization of inlier matches of the estimated homography. *Green arrows* correspond to SURF descriptor matches, and *red* ones are from dense optical flow (Color figure online)

balanced distribution of matched points, which is critical for a good homography estimation.

We, then, estimate the homography using the random sample consensus method (RANSAC; Fischler and Bolles 1981). RANSAC is a robust, non-deterministic algorithm for estimating the parameters of a model. At each iteration it randomly samples a subset of the data to estimate the parameters of the model and computes the number of inliers that fit the model. The final estimated parameters are those with the greatest consensus. We then rectify the image using the homography to remove the camera motion. Figure 1 (two columns in the middle) demonstrates the difference of optical flow before and after rectification. Compared to the original flow (the second column), the rectified version (the third column) suppresses the background camera motion and enhances the foreground moving objects.

For trajectory features, there are two major advantages of canceling out camera motion from optical flow. First, the motion descriptors can directly benefit from this. As shown in Wang et al. (2013a), the performance of the HOF descriptor degrades significantly in the presence of camera motion. Our experimental results in Sect. 7.1 show that HOF can achieve similar performance as MBH when we have corrected the optical flow. The combination of HOF and MBH can further improve the results as they represent zero-order (HOF) and first-order (MBH) motion information.

Second, we can remove trajectories generated by camera motion. This can be achieved by thresholding the displacement vectors of the trajectories in the warped flow field. If the displacement is very small, the trajectory is considered to be too similar to camera motion, and thus removed. Figure 3 shows examples of removed background trajectories. Our method works well under various camera motions (such as pan, tilt and zoom) and only trajectories related to human actions are kept (shown in green in Fig. 3). This gives us similar effects as sampling features based on visual saliency maps (Mathe and Sminchisescu 2012; Vig et al. 2012).

The last column of Fig. 3 shows two failure cases. The top one is due to severe motion blur, which makes both SURF descriptor matching and optical flow estimation unreliable. Improving motion estimation in the presence of motion blur is worth further attention, since blur often occurs in realistic datasets. In the bottom example, humans dominate the frame, which causes homography estimation to fail. We discuss a solution for the latter case below.

3.3 Removing Inconsistent Matches Due to Humans

In action datasets, videos often focus on the humans performing the action. As a result, it is very common that humans dominate the frame, which can be a problem for camera motion estimation as human motion is in general not consistent with it. We propose to use a human detector to remove matches from human regions. In general, human detection in action datasets is rather difficult, as humans appear in many different poses when performing the action. Furthermore, the person could be only partially visible due to occlusion or being partially out of view.

Here, we apply a state-of-the-art human detector (Prest et al. 2012), which adapts the general part-based human detector (Felzenszwalb et al. 2010) to action datasets. The detector combines several part detectors dedicated to different regions of the human body (including full person, upper-body and face). It is trained using the PASCAL VOC07 training data for humans as well as near-frontal upper-bodies from (Ferrari et al. 2008). We set the detection threshold to 0.1. If the confidence of a detected window is higher than that, we consider it to be a positive sample. This is a high-recall operating point where few human detections are

Fig. 3 Examples of removed trajectories under various camera motions, eg, pan, zoom, tilt. *White trajectories* are considered due to camera motion. The *red dots* are the feature point positions in the current frame. The *last column* shows two failure cases. The *top* one is due to severe motion blur. The *bottom* one fits the homography to the moving humans as they dominate the whole frame

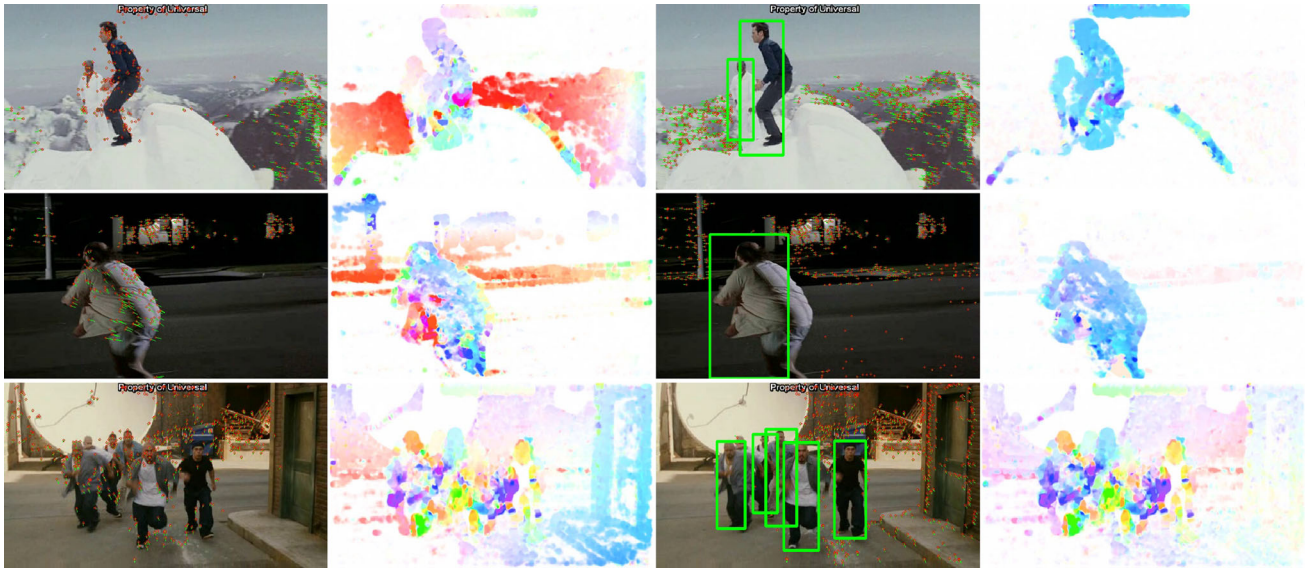

Fig. 4 Homography estimation without human detector (*left*) and with human detector (*right*). We show inlier matches in the *first* and *third columns*. The optical flow (*second* and *fourth columns*) is warped with the corresponding homography. The *first* and *second rows* show a clear improvement of the estimated homography when using a human detector. The *last row* presents a failure case. See the text for details

missed. Figure 4, third column, shows some examples of human detection results.

We use the human detector as a mask to remove feature matches inside the bounding boxes when estimating the homography. Without human detection (the left two columns of Fig. 4), many features from the moving humans become inlier matches and the homography is, thus, incorrect. As a result, the corresponding optical flow is not correctly warped. In contrast, camera motion is successfully compensated (the right two columns of Fig. 4), when the human bounding boxes are used to remove matches not corresponding to camera motion. The last row of Fig. 4 shows a failure case. The homography does not fit the background very well despite detecting the humans correctly, as the background is represented by two planes, one of which is very close to the camera. In our experiments we compare the performance with and without human detection.

The human detector does not always work perfectly. In Fig. 5, we show some failure cases, which are typically due to complex human body poses, self occlusion, motion blur etc. In order to compensate for missing detections, we track all the bounding boxes obtained by the human detector. Tracking is performed forward and backward for each frame of the video. Our approach is simple: we take the average motion vector (Farnebäck 2003) and propagate the detections to the next frame. We track each bounding box for at most 15 frames and stop if there is a 50 % overlap with another bounding box. All the human bounding boxes are available online.[2] In the following, we always use the human detector to remove potentially inconsistent matches before computing the homography, unless stated otherwise.

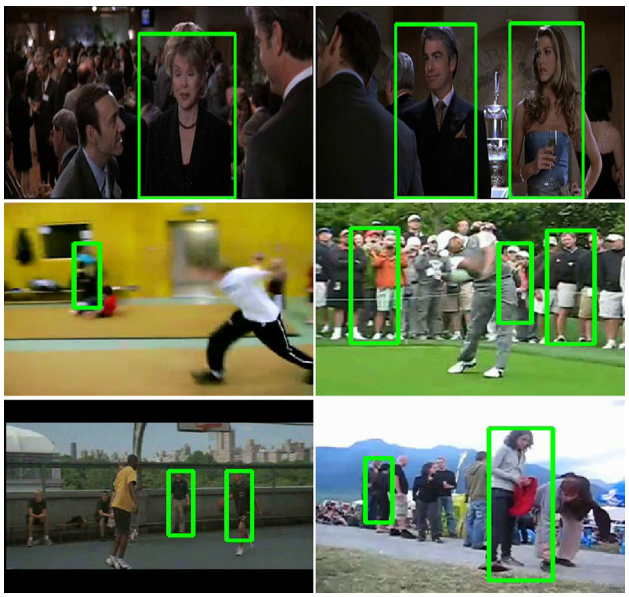

Fig. 5 Examples of human detection results. The *first row* is from Hollywood2, whereas the *last two rows* are from HMDB51. Not all humans are detected correctly as human detection on action datasets is very challenging

3.4 Improved Trajectory Features

To extract our improved trajectories, we sample and track feature points exactly the same way as in Wang et al. (2013a), see Sect. 3.1. To compute the descriptors, we first estimate

[2] http://lear.inrialpes.fr/~wang/improved_trajectories.

the homography with RANSAC using the feature matches extracted between each pair of consecutive frames; matches on detected humans are removed. We warp the second frame with the estimated homography. Homography estimation takes around 5 ms for each pair of frames. The optical flow (Farnebäck 2003) is then re-computed between the first and the warped second frame. Motion descriptors (HOF and MBH) are computed on the warped optical flow. The HOG descriptor remains unchanged. We estimate the homography and warped optical flow for every two frames independently to avoid error propagation. We use the same parameters and the RootSIFT normalization as the baseline described in Sect. 3.1. We further utilize these stabilized motion vectors to remove background trajectories. For each trajectory, we compute the maximal magnitude of the motion vectors during its length of 15 frames. If the maximal magnitude is lower than a threshold (set to one pixel, i.e., the motion displacement is less than one pixel between each pair of frames), the trajectory is considered to be consistent with camera motion, and thus removed.

4 Feature Encoding

In this section, we present how we aggregate local descriptors into a holistic representation, and augment this representation with weak spatio–temporal location information.

4.1 Fisher Vector

The FV (Sánchez et al. 2013) was found to be the most effective encoding technique in a recent evaluation study of feature pooling techniques for object recognition (Chatfield et al. 2011); this evaluation included also BOW, sparse coding techniques, and several variants. The FV extends the BOW representation as it encodes both first and second order statistics between the video descriptors and a diagonal covariance Gaussian mixture model (GMM). Given a video, let $x_n \in \mathbb{R}^D$ denote the nth D-dimensional video descriptor, q_{nk} the soft assignment of x_n to the kth Gaussian, and π_k, μ_k and σ_k are the weight, mean, and diagonal of the covariance matrix of the kth Gaussian respectively. After normalization with the inverse Fisher information matrix (which renders the FV invariant to the parametrization), the D-dimensional gradients w.r.t.the mean and variance of the kth Gaussian are given by:

$$G_{\mu_k} = \sum_{n=1}^{N} q_{nk} \left[x_n - \mu_k\right] / \sqrt{\sigma_k \pi_k}, \tag{1}$$

$$G_{\sigma_k} = \sum_{n=1}^{N} q_{nk} \left[(x_n - \mu_k)^2 - \sigma_k^2\right] / \sqrt{2\sigma_k^2 \pi_k}. \tag{2}$$

For each descriptor type x_n, we can represent the video as a $2DK$ dimensional Fisher vector. To compute FV, we first reduce the descriptor dimensionality by a factor of two using principal component analysis (PCA), as in Sánchez et al. (2013). We then randomly sample a subset of $1000 \times K$ descriptors from the training set to estimate a GMM with K Gaussians. After encoding the descriptors using Eqs. (1) and (2), we apply power and ℓ_2 normalization to the final Fisher vector representation as in Sánchez et al. (2013). A linear SVM is used for classification.

Besides FV, we also consider BOW histograms as a baseline for feature encoding. We use the soft assignments to the same Gaussians as used for the FV instead of hard assignment with k-means clustering (van Gemert et al. 2010). Soft assignments have been reported to yield better performance, and since the same GMM vocabulary is used as for the FV, it also rules out any differences due to the vocabulary. For BOW, we consider both linear and RBF-χ^2 kernel for the SVM classifier. In the case of linear kernel, we employ the same power and ℓ_2 normalization as FV, whereas ℓ_1 normalization is used for RBF-χ^2 kernel.

To combine different descriptor types, we encode each descriptor type separately and concatenate their normalized BOW or FV representations together. In the case of multi-class classification, we use a one-against-rest approach and select the class with the highest score. For the SVM hyperparameters, we set the class weight w to be inversely proportional to the number of samples in each class so that both positive and negative classes contribute equally in the loss function. We set the regularization parameter C by cross validation on the training set, by testing values in the range $C \in \{3^{-2}, 3^{-1}, \ldots, 3^{7}\}$. In all experiments, we use the same settings.

4.2 Weak Spatio–Temporal Location Information

To go beyond a completely orderless representation of the video content in a BOW histogram or FV, we consider including a weak notion of spatio–temporal location information of the local features. For this purpose, we use the spatio–temporal pyramid (STP) representation (Laptev et al. 2008), and compute separate BOW or FV over cells in spatio–temporal grids. We also consider the spatial Fisher vector (SFV) of Krapac et al. (2011), 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 feature vectors (HOG, HOF or MBH) with the 3D locations, as done in McCann and Lowe (2013) and Sánchez et al. (2012); the main difference being that the latter do clustering on the extended feature vectors while this is not the case for the SFV. SFV is also computed in each cell of STP. To combine SFV with BOW or FV, we simply concatenate them together.

5 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, see Fig. 6. This is due to the fact that if a relatively long action appears, it is likely that there are short sub-sequences 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 as well (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 entire video, (ii) does not have overlapping windows, and (iii) maximizes the sum of scores of the selected windows. We formally express this method as an optimization problem:

$$\begin{aligned} \underset{y}{\text{maximize}} \quad & \sum_{i=1}^{n} y_i s_i \\ \text{subject to} \quad & \bigcup_{i:y_i=1} l_i = T, \\ & \forall_{y_i=y_j=1} : l_i \cap l_j = \emptyset, \\ & y_i \in \{0,1\},\ i = 1,\ldots,n. \end{aligned} \tag{3}$$

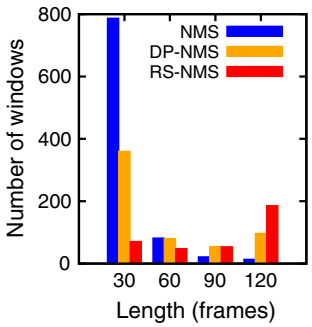

Fig. 6 Histograms of the window sizes on the *Coffee and Cigarettes* dataset after three variants of non-maxima suppression: classic non-maximum suppression (NMS), dynamic programming non-maximum suppression (DP-NMS), and re-scored non-maximum suppression (RS-NMS). Two of the methods, NMS and DP-NMS, select mostly short windows, 30-frames long, while the RS-NMS variant sets a bias towards longer windows, 120-frames long. In practice we prefer longer windows as they tend to cover better the action

where the boolean variables $y_1, \ldots, y_n$ represent the subset; s_i and l_i denote the score and the interval of window i; n is the total number of windows; T is the interval that spans the whole video.

The optimal subset is found efficiently by dynamic programming as follows. We first divide the temporal domain into discrete time steps. With each time step we associate a latent state: the temporal window that contains that particular time step. Each window is characterized by its starting point and duration. A pairwise potential is used to enforce the first two constraints (full duration coverage and non-overlapping segments): if a segment is not terminated at the current time step, the next time step should still be covered by the current segment, otherwise a new segment should be started. We maximize the score based on an unary potential that is defined as the score of the associated time step. The dynamic programming Viterbi algorithm is used to compute the optimal solution for the optimization problem of Eq. (3) using a forwards and backwards pass over the time steps. The runtime is linear in the number of time steps. We refer to this method as DP-NMS.

Figure 6 shows the histogram of durations of the windows that pass the non-maximum suppression stage using the different techniques, for the action *smoking* used in our experiments in Sect. 7.2. The durations for the two proposed methods, DP-NMS and RS-NMS, have a more uniform distribution than that for the standard NMS method, with RS-NMS favouring the longest windows. This behaviour is also observed in Fig. 7, which gives an example of the different windows retained for a specific video segment of the *Coffee & Cigarettes* movie. DP-NMS selects longer windows than NMS, but they do not align well with the action and the score of the segments outside the action are high. For this example, RS-NMS gives the best selection among the three methods, as it retains few segments and covers the action accurately.

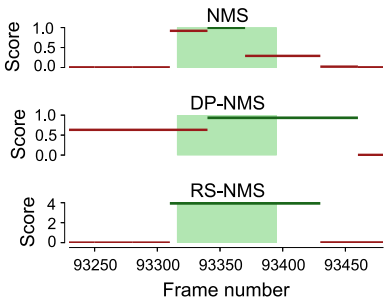

Fig. 7 Windows retained by NMS variants, green if they overlap more than 20 % with the true positive, red otherwise. The green region denotes the ground-truth action. For the NMS, the segments selected are too short. The DP-NMS selects longer segments, but it does not align well with the true action as it maximizes the total score over the whole video. The RS-NMS strikes a good balance of the segment's length and their score, and it gives the best solution in this example

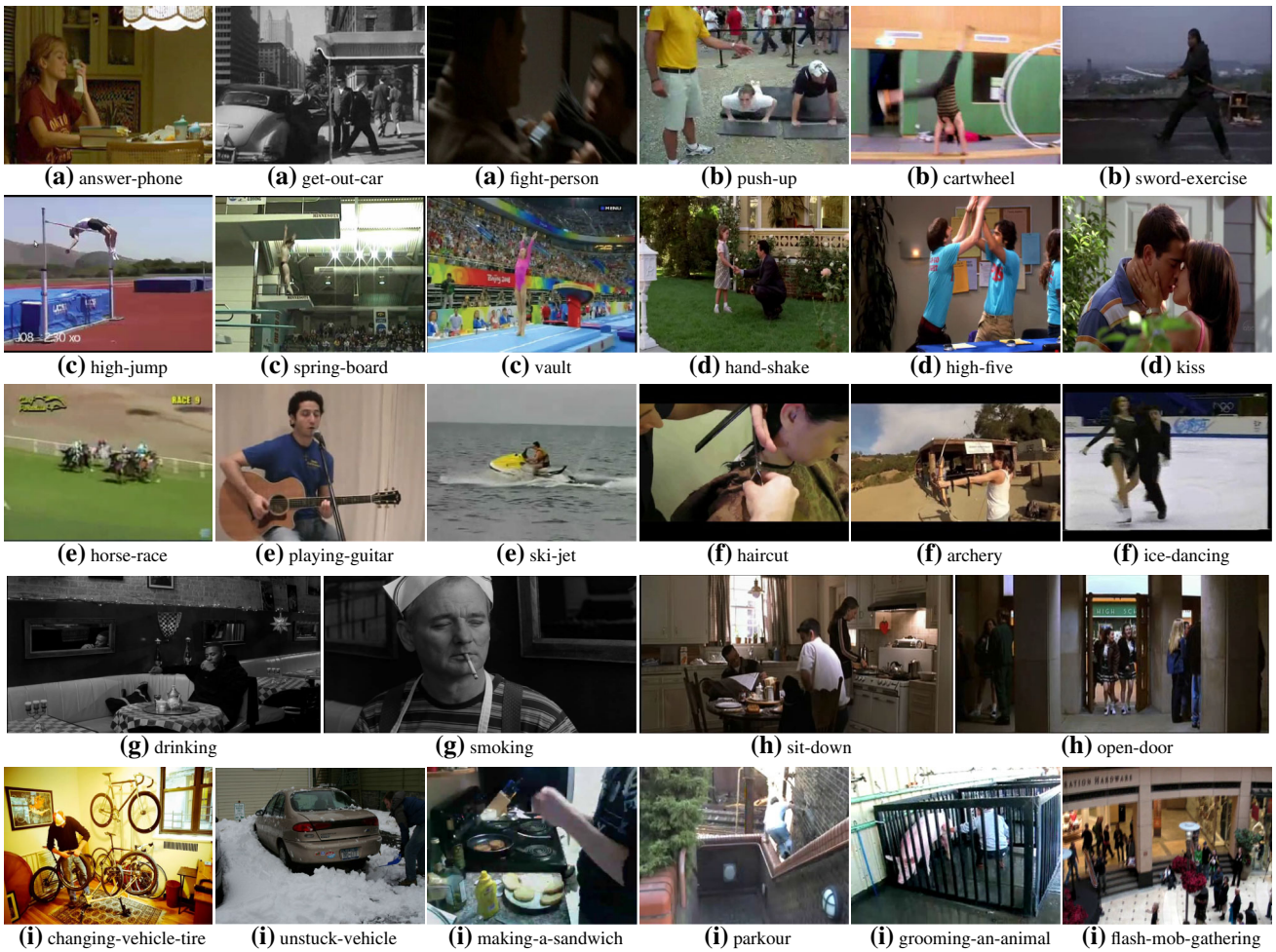

Fig. 8 From *top* to *bottom*, example frames from **a** Hollywood2, **b** HMDB51, **c** Olympic Sports, **d** High Five, **e** UCF50, **f** UCF101, **g** *Coffee and Cigarettes*, **h** DLSBP and **i** TRECVID MED 2011

6 Datasets Used for Experimental Evaluation

In this section, we briefly describe the datasets and their evaluation protocols for the three tasks. We use six challenging datasets for action recognition (i.e., Hollywood2, HMDB51, Olympic Sports, High Five, UCF50 and UCF101), *Coffee and Cigarettes* and DLSBP for action detection, and TRECVID MED 2011 for large scale event detection. In Fig. 8, we show some sample frames from the datasets.

6.1 Action Recognition

The **Hollywood2** dataset (Marszałek et al. 2009) has been collected from 69 different Hollywood movies and includes 12 action classes. It contains 1707 videos split into a training set (823 videos) and a test set (884 videos). Training and test videos come from different movies. The performance is measured by mean average precision (mAP) over all classes, as in Marszałek et al. (2009).

The **HMDB51** dataset (Kuehne et al. 2011) is collected from a variety of sources ranging from digitized movies to YouTube videos. In total, there are 51 action categories and 6766 video sequences. We follow the original protocol using three train-test splits (Kuehne et al. 2011). For every class and split, there are 70 videos for training and 30 videos for testing. We report average accuracy over the three splits as performance measure. Note that in all the experiments we use the original videos, not the stabilized ones.

The **Olympic Sports** dataset (Niebles et al. 2010) consists of athletes practicing different sports, which are collected from YouTube and annotated using Amazon Mechanical Turk. There are 16 sports actions (such as high-jump, pole-vault, basketball lay-up, discus), represented by a total of 783 video sequences. We use 649 sequences for training and 134 sequences for testing as recommended by the authors. We report mAP over all classes, as in Niebles et al. (2010).

The **High Five** dataset (Patron-Perez et al. 2010) consists of 300 video clips extracted from 23 different TV shows.

Each of the clips contains one of four interactions: hand shake, High Five, hug and kiss (50 videos for each class). Negative examples (clips that don't contain any of the interactions) make up the remaining 100 videos. Though the dataset is relatively small, it is challenging due to large intra-class variation, and all the action classes are very similar to each other (i.e., interactions between two persons). We follow the original setting in Patron-Perez et al. (2010), and compute average precision (AP) using a pre-defined two-fold cross-validation.

The **UCF50** dataset (Reddy and Shah 2012) has 50 action categories, consisting of real-world videos taken from YouTube. The actions range from general sports to daily life exercises. For all 50 categories, the videos are split into 25 groups. For each group, there are at least four action clips. In total, there are 6618 video clips. The video clips in the same group may share some common features, such as the same person, similar background or viewpoint. We apply the leave-one-group-out cross-validation as recommended in Reddy and Shah (2012) and report average accuracy over all classes.

The **UCF101** dataset (Soomro et al. 2012) is extended from UCF50 with additional 51 action categories. In total, there are 13,320 video clips. We follow the evaluation guidline from the THUMOS'13 workshop (Jiang et al. 2013) using three train-test splits. In each split, clips from seven of the 25 groups are used as test samples, and the rest for training. We report average accuracy over the three splits as performance measure.

6.2 Action Localization

The first dataset for action localization is extracted from the movie **Coffee and Cigarettes**, and contains annotations for the actions *drinking* and *smoking* (Laptev and Pérez 2007). The training set contains 41 and 70 examples for each class respectively. Additional training examples (32 and eight 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. (Duchenne et al. 2009) 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 AP score as in (Duchenne et al. 2009; Gaidon et al. 2011; Kläser et al. 2010; Laptev and Pérez 2007); considering a detection as correct when it overlaps (as measured by intersection over union) by at least 20 % with a ground truth annotation.

6.3 Event Recognition

The **TRECVID MED 2011** dataset (Over et al. 2012) is the largest dataset we consider. It consists of consumer videos from 15 categories that are more complex than the basic actions considered in the other datasets, eg, *changing a vehicle tire*, or *birthday party*. For each category between 100 and 300 training videos are available. In addition, 9600 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 over 1000 h, 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. The performance is evaluated using AP measure. The second setup is the one of Tang et al. (Tang et al. 2012). They split the data into three subsets: EVENTS, which contains 2048 videos from the 15 categories, but doesn't include the *null* class; DEV-T, which contains 602 videos from the first five categories and the 9,600 *null* videos; and DEV-O, which is the standard test set of 32,000 videos.[3] As in (Tang et al. 2012), we train on the EVENTS set and report the performance in AP on the DEV-T set for the first five categories and on the DEV-O set for the remaining ten actions.

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 128×88 to full HD 1920×1080. We rescale the videos to a width of at most 480 pixels, preserving the aspect ratio, and temporally subsample them by discarding every second frame in order to make the dataset computationally more tractable. These rescaling parameters were selected on a subset of the MED dataset; we present an exhaustive evaluation of the impact of the video resolution in Sect. 7.3. Finally, we also randomly sample the generated features to reduce the computational cost for feature encoding. This is done only for videos longer than 2000 frames, i.e., the sampling ratio is set to 2000 divided by the total number of frames.

[3] The number of videos in each subset varies slightly from the figures reported in (Tang et al. 2012). The reason is that there are multiple releases of the data. For our experiments, we used the labels from the `LDC2011E42` release.

Table 1 Comparison of bag-of-words and Fisher vectors using the non-stabilized MBH descriptor under different parameter settings

K	STP	Hollywood2					HMDB51				
		Bag-of-words			Fisher vector		Bag-of-words			Fisher vector	
		χ^2 Kernel	Linear kernel		Linear kernel		χ^2 Kernel	Linear kernel		Linear kernel	
		BOW (%)	BOW (%)	BOW+SFV (%)	FV (%)	FV+SFV (%)	BOW (%)	BOW (%)	BOW+SFV (%)	FV (%)	FV+SFV (%)
64	–	44.4	39.8	40.3	55.0	56.5	30.5	28.3	28.0	45.8	47.9
64	H3	48.0	44.9	45.0	57.9	59.2	35.8	30.1	33.1	48.0	49.4
64	T2	48.3	43.4	46.8	57.1	58.5	34.9	30.9	32.5	48.3	49.5
64	T2 + H3	50.2	46.8	46.4	59.4	59.5	37.1	32.5	34.2	50.3	51.1
128	–	45.8	42.1	43.5	57.1	58.5	33.8	31.9	32.2	48.2	50.3
128	H3	51.3	46.2	48.1	58.8	60.0	38.0	32.3	37.5	49.9	51.1
128	T2	50.5	45.5	49.4	58.8	59.9	38.2	32.9	36.2	50.2	51.1
128	T2 + H3	52.4	48.4	48.2	61.0	60.7	40.5	35.8	37.9	51.9	52.6
256	–	49.4	44.9	45.9	57.9	59.6	36.6	33.1	35.0	50.0	51.9
256	H3	52.9	46.0	50.6	59.0	61.0	40.6	36.2	40.4	51.4	52.3
256	T2	52.0	47.0	51.3	59.3	60.3	41.3	35.7	39.7	51.5	52.0
256	T2 + H3	53.6	50.2	50.2	61.0	61.3	43.5	39.2	41.2	52.6	53.2
512	–	50.2	46.8	49.0	58.9	60.5	40.3	35.6	37.9	51.3	53.2
512	H3	53.1	49.5	51.2	59.5	61.5	43.4	38.4	41.5	51.4	52.3
512	T2	53.9	49.4	52.8	60.2	61.0	42.6	39.1	42.2	52.2	53.3
512	T2 + H3	55.5	51.6	51.3	**61.7**	**61.9**	45.2	42.1	43.5	52.7	53.7
1024	–	52.3	48.5	50.4	58.9	60.9	42.3	39.2	39.9	51.4	53.9
1024	H3	55.6	50.6	52.6	59.4	61.2	45.4	40.8	44.2	51.7	52.8
1024	T2	54.6	52.0	**54.5**	59.7	60.7	46.0	41.8	**46.3**	52.5	53.0
1024	T2 + H3	**56.6**	**52.9**	53.5	61.2	61.8	**47.5**	**43.9**	45.7	**53.3**	**53.8**

We use ℓ_1 normalization for the χ^2 kernel, and power and ℓ_2 normalization for the linear kernel
The best results for each setting are shown in bold

7 Experimental Results

Below, we present our experimental evaluation results for action recognition in Sect. 7.1, for action localization in Sect. 7.2, and for event recognition in Sect. 7.3.

7.1 Action Recognition

We first compare BOW and FV for feature encoding, and evaluate the performance gain due to different motion stabilization steps. Then, we assess the impact of removing inconsistent matches based on human detection, and finally compare to the state of the art.

7.1.1 Feature encoding with BOW and FV

We begin our experiments with the original non-stabilized MBH descriptor (Wang et al. 2013a) and compare its performance using BOW and FV under different parameter settings. For this initial set of experiments, we chose the Hollywood2 and HMDB51 datasets as they are widely used and are representative in difficulty and size for the task of action recognition. We evaluate the effect of including weak geometric information using the spatial Fisher vector (SFV) and STP. We consider STP grids that divide the video in two temporal parts (T2), and/or three spatial horizontal parts (H3). When using STP, we always concatenate the representations (i.e., BOW or FV) over the whole video. For the case of T2 + H3, we concatenate all six BOW or FV representations (one for the whole video, two for T2, and three for H3). Unlike STP, the SFV has only a limited effect for FV on the representation size, as it just adds six dimensions (for the spatio–temporal means and variances) for each visual word. For the BOW representation, the situation is 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 seven; similar to the factor six for STP.

Table 1 lists all the results using different settings on Hollywood2 and HMDB51. It is obvious that increasing the number of Gaussians K leads to significant performance gain for both BOW and FV. However, the performance of FV tends to saturate after $K = 256$, whereas BOW keeps improving up

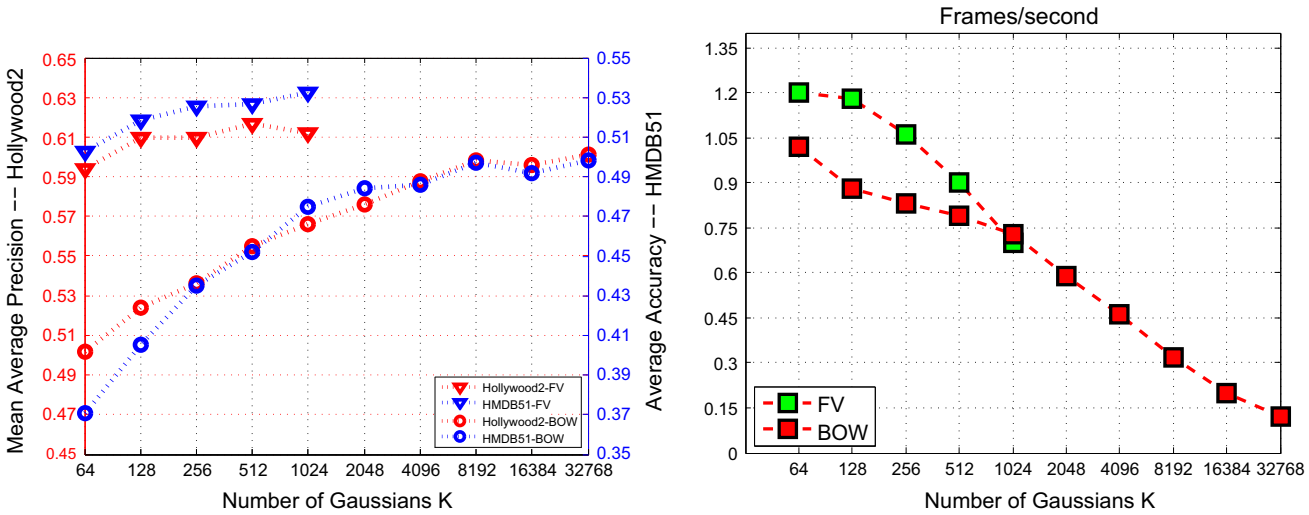

Fig. 9 Comparing BOW (RBF-χ^2 kernel) using large vocabularies with FV (linear kernel). For both, we only use STP (T2 + H3) without SFV. *Left* performance on Hollywood2 and HMDB51. *Right* runtime speed on a Hollywood2 video of resolution 720 × 480 pixels

to $K = 1024$. This is probably due to the high dimensionality of FV which results in an earlier saturation. Both BOW and FV benefit from including STP and SFV, which are complementary since the best performance is always obtained when they are combined.

As expected, the RBF-χ^2 kernel works better than the linear kernel for BOW. Typically, the difference is around 4–5 % on both Hollywood2 and HMDB51. When comparing different feature encoding strategies, the FV usually outperforms BOW by 6–7 % when using the same number of visual words. Note that FV of 64 visual words is even better than BOW of 1024 visual words; confirming that for FV fewer visual words are needed than for BOW.

We further explore the limits of BOW performance by using very large vocabularies, i.e., with K up to 32, 768. The results are shown in the left panel of Fig. 9. For BOW, we use χ^2 kernel and T2 + H3 which give the best results in Table 1. For a fair comparison, we only use T2 + H3 for FV without SFV. On both Hollywood2 and HMDB51, the performance of BOW becomes saturated when K is larger than 8192. If we compare BOW and FV representations with similar dimensions (i.e., $K = 32,768$ for BOW and K between 64 and 128 for FV), FV still outperforms BOW by 2 % on HMDB51 and both have comparable performance for Hollywood2. Moreover, feature encoding with large vocabularies is very time-consuming as shown in the right panel of Fig. 9, where $K = 32,768$ for BOW is eight times slower than $K = 128$ for FV. This can impose huge computational cost for large datasets such as TRECVID MED. FV is also advantageous as it achieves excellent results with a linear SVM which is more efficient than kernel SVMs. Note however, that the classifier training time is negligible compared to the feature extraction and encoding time, eg, it only takes around 200 s for FV with $K = 256$ to compute the Gram matrix and to train the classifiers on the Hollywood2 dataset.

To sum up, we choose FV with both STP and SFV, and set $K = 256$ for a good compromise between accuracy and computational complexity. We use this setting in the rest of experiments unless stated otherwise.

7.1.2 Evaluation of Improved Trajectory Features

We choose the dense trajectories (Wang et al. 2013a) as our baseline, compute HOG, HOF and MBH descriptors as described in Sect. 3.1, and report results on all the combinations of them. In order to evaluate intermediate results, we decouple our method into two parts, i.e., "WarpFlow" and "RmTrack", which stand for warping optical flow with the homography and removing background trajectories consistent with the homography. The combined setting uses both. The results are presented in Table 2 for Hollywood2 and HMDB51.

In the following, we discuss the results per descriptor. The results of HOG are similar for different variants on both datasets. Since HOG is designed to capture static appearance information, we do not expect that compensating camera motion significantly improves its performance.

HOF benefits the most from stabilizing optical flow. Both "Combined" and "WarpFlow" are substantially better than the other two. On Hollywood2, the improvements are around 5 %. On HMDB51, the improvements are even higher: around 10 %. After motion compensation, the performance of HOF is comparable to that of MBH.

MBH is known for its robustness to camera motion (Wang et al. 2013a). However, its performance still improves, as

Table 2 Comparison of baseline to our method and intermediate

	Hollywood2				HMDB51			
	Baseline (%)	WarpFlow (%)	RmTrack (%)	Combined (%)	Baseline (%)	WarpFlow (%)	RmTrack (%)	Combined (%)
HOG	51.3	52.1	52.6	53.0	42.0	43.1	44.7	44.4
HOF	56.4	61.5	57.6	62.4	43.3	51.7	45.3	52.3
MBH	61.3	63.1	63.1	63.6	53.2	55.3	55.9	56.9
HOG + HOF	61.9	64.3	63.2	65.3	51.9	56.5	54.2	57.5
HOG + MBH	63.0	64.2	63.6	64.7	**56.3**	57.8	57.7	58.7
HOF + MBH	62.0	65.3	62.7	65.2	53.2	57.1	54.8	58.3
HOG + HOF + MBH	**63.6**	**65.7**	**65.0**	**66.8**	55.9	**59.6**	**57.8**	**60.1**

WarpFlow: computing HOF and MBH using warped optical flow, while keeping all the trajectories. RmTrack: removing background trajectories, but compute descriptors using the original flow. Combined: removing background trajectories, and descriptors on warped flow. All the results use SFV + STP, $K = 256$, and human detection to remove outlier matches
The best results for each setting are shown in bold

Table 3 Impact of human detection on a subset of Hollywood2 and High Five datasets

	Hollywood2-sub				High Five			
	Baseline (%)	Non (%)	Automatic (%)	Manual (%)	Baseline (%)	Non (%)	Automatic (%)	Manual (%)
HOG	39.9	40.0	39.7	40.4	48.2	49.7	49.3	50.2
HOF	40.7	49.6	51.5	52.1	53.4	66.8	67.4	68.1
MBH	49.6	52.5	53.1	54.2	61.5	67.3	68.5	68.8
HOG + HOF	46.3	49.9	51.3	52.8	57.5	66.3	67.5	67.5
HOG + MBH	49.8	51.5	52.3	53.4	61.8	66.9	67.2	67.8
HOF + MBH	49.6	53.8	54.4	55.3	61.4	**69.1**	**70.5**	**71.2**
HOG + HOF + MBH	**50.8**	**54.3**	**55.5**	**56.3**	**62.5**	68.1	69.4	69.8

Baseline without motion stabilization, *Non* without human detection, *Automatic* automatic human detection, *Manual* manually annotation. As before, we use SFV + STP, and set $K = 256$
The best results for each setting are shown in bold

motion boundaries are much clearer, see Figs. 1 and 4. We have over 2 % improvement on both datasets.

Combining HOF and MBH further improves the results as they are complementary to each other. HOF represents zero-order motion information, whereas MBH focuses on first-order derivatives. Combining all three descriptors achieve the best performance, as shown in the last row of Table 2.

7.1.3 Removing Inconsistent Matches Due to Humans

We investigate the impact of removing inconsistent matches due to humans when estimating the homography, see Fig. 4 for an illustration. We compare four cases: (i) the baseline without stabilization, (ii) estimating the homography without human detection, (iii) with automatic human detection, and (iv) with manual labeling of humans. This allows us to measure the impact of removing matches from human regions as well as to determine an upper bound in case of a perfect human detector. We consider two datasets: Hollywood2 and High Five. To limit the labeling effort on Hollywood2, we annotated humans in 20 training and 20 testing videos for each action class. On High Five, we use the annotations provided by the authors of Patron-Perez et al. (2010).

As shown in Table 3, human detection helps to improve motion descriptors (i.e., HOF and MBH), since removing inconsistent matches on humans improves the homography estimation. Typically, the improvements are over 1 % when using an automatic human detector or manual labeling. The last two rows of Table 4 show the impact of automatic human detection on all six datasets. Human detection always improves the performance slightly.

7.1.4 Comparison to the State of the Art

Table 4 compares our method with the most recent results reported in the literature. On Hollywood2, all presented results (Jain et al. 2013; Jiang et al. 2012; Mathe and Sminchisescu 2012; Zhu et al. 2013) improve dense trajectories in different ways. Mathe and Sminchisescu (2012) prune background features based on visual saliency. Zhu et al. (2013)

Table 4 Comparison of our results (HOG + HOF + MBH) to the state of art

Hollywood2 (%)		HMDB51 (%)		Olympic Sports (%)	
Jiang et al. (2012)	59.5	Jiang et al. (2012)	40.7	Jain et al. (2013)	83.2
Mathe and Sminchisescu (2012)	61.0	Ballas et al. (2013)	51.8	Li et al. (2013)	84.5
Zhu et al. (2013)	61.4	Jain et al. (2013)	52.1	Wang et al. (2013b)	84.9
Jain et al. (2013)	62.5	Zhu et al. (2013)	54.0	Gaidon et al. (2013)	85.0
Baseline	63.6	Baseline	55.9	Baseline	85.8
Without HD	66.1	Without HD	59.3	Without HD	89.6
With HD	**66.8**	With HD	**60.1**	With HD	**90.4**
High Five (%)		UCF50 (%)		UCF101 (%)	
Ma et al. (2013)	53.3	Shi et al. (2013)	83.3	Peng et al. (2013)	84.2
Yu et al. (2012)	56.0	Wang et al. (2013b)	85.7	Murthy and Goecke (2013a)	85.4
Gaidon et al. (2013)	62.4	Ballas et al. (2013)	**92.8**	Karaman et al. (2013)	85.7
Baseline	62.5	Baseline	89.1	Baseline	83.5
Without HD	68.1	Without HD	91.3	Without HD	85.7
With HD	**69.4**	With HD	91.7	With HD	**86.0**

We present our results for FV encoding ($K = 256$) using SFV + STP both with and without automatic human detection (HD). Best result for each dataset is marked in bold
The best result for each dataset shown in bold

apply multiple instance learning on top of dense trajectory features in order to learn mid-level "acton" to better represent human actions. Recently, Jain et al. (2013) report 62.5 % by decomposing visual motion to stabilize dense trajectories. We further improve their results by over 4 %.

HMDB51 (Kuehne et al. 2011) is a relatively new dataset. Jiang et al. (2012) achieve 40.7 % by modeling the relationship between dense trajectory clusters. Ballas et al. (2013) report 51.8 % by pooling dense trajectory features from regions of interest using video structural cues estimated by different saliency functions. The best previous result is from (Zhu et al. 2013). We improve it further by over 5 %, and obtain 60.1 % accuracy.

Olympic Sports (Niebles et al. 2010) contains significant camera motion, which results in a large number of trajectories in the background. Li et al. (2013) report 84.5 % by dynamically pooling feature from the most informative segments of the video. Wang et al. (2013b) propose motion atom and phrase as a mid-level temporal part for representing and classifying complex action, and achieve 84.9 %. Gaidon et al. (2013) model the motion hierarchies of dense trajectories (Wang et al. 2013a) with tree structures and report 85.0 %. Our improved trajectory features outperform them by over 5 %.

High Five (Patron-Perez et al. 2010) focuses on human interactions and serves as a good testbed for various structure model applied for action recognition. Ma et al. (2013) propose hierarchical space-time segments as a new representation for simultaneously action recognition and localization. They only extract the MBH descriptor from each segment and report 53.3 % as the final performance. Yu et al. (2012) propagate Hough voting of STIP (Laptev et al. 2008) features in order to overcome their sparseness, and achieve 56.0 %. With our framework we achieve 69.4 % on this challenging dataset.

UCF50 (Reddy and Shah 2012) can be considered as an extension of the widely used YouTube dataset (Liu et al. 2009). Recently, Shi et al. (2013) report 83.3 % using randomly sampled HOG, HOF, HOG3D and MBH descriptors. Wang et al. (2013b) achieve 85.7 %. The best result so far is 92.8 % from Ballas et al. (2013). We obtain a similar accuracy of 91.7 %.

UCF101 (Soomro et al. 2012) is used in the recent THUMOS'13 Action Recognition Challenge (Jiang et al. 2013). All the top results are built on different variants of dense trajectory features (Wang et al. 2013a). Karaman et al. (2013) extract many features (such as HOG, HOF, MBH, STIP, SIFT, etc.) and do late fusion with logistic regression to combine the output of each feature channel. Murthy and Goecke (2013a) combine ordered trajectories (Murthy and Goecke 2013b) and improved trajectories (Wang and Schmid 2013), and apply Fisher vector to encode them. With our framework we obtained 86.0 %, and ranked first among all 16 participants.

7.2 Action Localization

In our second set of experiments we consider the localization of four actions (i.e., *drinking*, *smoking*, *open door* and *sit down*) in feature length movies. We set the encoding parameters the same as action recognition: $K = 256$ for Fisher vector with SFV + STP. We first consider the effect of dif-

Table 5 Evaluation of the non-maximum suppression variants: classic non-maximum suppression (NMS), dynamic programming non-maximum suppression (DP-NMS), and re-scored non-maximum suppression (RS-NMS)

	Overlap	Drinking (%)	Smoking (%)	Open door (%)	Sit down (%)
NMS	20	73.2	32.3	23.3	28.6
RS-NMS	20	76.5	38.0	23.2	26.6
DP-NMS	0	71.4	36.7	21.0	23.6
NMS	0	74.1	32.4	24.2	**28.9**
RS-NMS	0	**80.2**	**40.9**	**26.0**	27.1

The overlap parameter (second column) indicates the maximum overlap (intersection over union) allowed between any two windows after non-maximum suppression. We use HOG + HOF + MBH from improved trajectory features (without human detector) with FV ($K = 256$) augmented by SFV + STP
The best results for each class are shown in bold

Table 6 Comparison of improved trajectory features (with and without human detection) to the baseline for the action localization task

	Drinking			Smoking		
	Baseline (%)	Without HD (%)	With HD (%)	Baseline (%)	Without HD (%)	With HD (%)
HOG	44.3	52.7	51.5	31.0	32.9	33.9
HOF	82.5	79.2	79.1	28.9	34.7	33.9
MBH	78.7	73.0	70.4	**47.7**	**48.7**	43.2
HOG + HOF	80.8	**81.1**	**79.9**	35.5	33.5	33.0
HOG + MBH	78.2	74.3	75.0	40.5	42.7	42.3
HOF + MBH	**85.0**	79.0	78.3	46.8	45.7	**45.0**
HOG + HOF + MBH	81.6	80.2	79.0	38.5	40.9	39.4
	Open door			Sit down		
	Baseline (%)	Without HD (%)	With HD (%)	Baseline (%)	Without HD (%)	With HD (%)
HOG	21.6	23.8	21.4	14.9	14.3	14.3
HOF	21.4	19.8	23.9	25.5	25.5	23.8
MBH	29.5	23.4	22.9	26.1	25.8	25.6
HOG + HOF	20.9	27.5	26.9	24.1	21.9	22.6
HOG + MBH	**29.6**	**30.2**	**29.2**	28.3	25.0	25.2
HOF + MBH	28.8	23.4	23.8	**30.6**	**27.2**	27.1
HOG + HOF + MBH	28.8	26.0	26.4	29.6	27.1	**27.6**

We use Fisher vector ($K = 256$) with SFV + STP to encode local descriptors, and apply RS-NMS-0 for non-maxima suppression. We show results on two datasets: the *Coffee & Cigarettes* dataset (Laptev and Pérez 2007) (*drinking* and *smoking*) and the DLSBP dataset (Duchenne et al. 2009) (*open door* and *sit down*)
The best results for each setting are shown in bold

ferent NMS variants using our improved trajectory features without human detection. We then compare with the baseline dense trajectory features and discuss the impact of human detection. Finally we present a comparison to the state-of-the-art methods.

7.2.1 Evalution of NMS Variants

We report all the results by combining HOF, HOF and MBH together, and present them in Table 5. We see that simple rescoring (RS-NMS) significantly improves over standard NMS on two out of four classes, while the dynamic programming version (DP-NMS) is slightly inferior when compared with RS-NMS. 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 all four action classes, while for RS-NMS zero overlap is beneficial on all classes. Since RS-NMS zero overlap performs the best among all five different variants, we use it in the remainder of the experiments.

7.2.2 Evaluation of Improved Trajectory Features

We present detailed experimental results in Table 6. We analyze all the combinations of the three descriptors and compare our improved trajectory features (with and without human detection) with the baseline dense trajectory features.

Table 7 Improved trajectory features without human detection compared to the state of the art for localization

	Drinking (%)	Smoking (%)	Open door (%)	Sit down (%)
Laptev and Pérez (2007)	49.0	–	–	–
Duchenne et al. (2009)	40.0	–	14.4	13.9
Kläser et al. (2010)	54.1	24.5	–	–
Gaidon et al. (2011)	57.0	31.0	16.4	19.8
RS-NMS zero overlap	**80.2**	**40.9**	**26.0**	**27.1**

We use HOG + HOF + MBH descriptors encoded with FV ($K = 256$) and SFV + STP, and apply RS-NMS zero overlap for non-maxima suppression
The best results for each class are shown in bold

We observe that combining all descriptors usually gives better performance than individual descriptors. The improved trajectory features are outperformed by the baseline on three out of four classes for the case of HOG + HOF + MBH. Note that the results of different descriptors and settings are less consistent than they are on action recognition datasets, eg, Table 2, as here we report the results for each class separately. Furthermore, since the action localization datasets are much smaller than action recognition ones, the number of positive examples per category is limited, which renders the experimental results less stable. In randomised experiments, where we leave one random positive test sample out from the test set, we observe standard deviations of the same order as the differences between the various settings (not shown for sake of brevity).

As for the impact of human detection, surprisingly leaving it out performs better for *drinking* and *smoking*. Since *Coffee & Cigarettes* essentially consists of scenes with static camera, this result might be due to inaccuracies in the homography estimation.

7.2.3 Comparison to the State of the Art

In Table 7, we compare our RS-NMS zero overlap method with previously reported state-of-the-art results. As features we use HOG + HOF + MBH of the improved trajectory features, but without human detection. We obtain substantial improvements on all four action classes, despite the fact that previous work used more elaborate techniques. For example, Kläser et al. (2010) relied on human detection and tracking, while Gaidon et al. (2011) requires finer annotations that indicate the position of characteristic moments of the actions (actoms). The biggest difference comes from the *drinking* class, where our result is over 23 % better than that of Gaidon et al. (2011).

7.3 Event Recognition

In our last set of experiments we consider the large-scale TRECVID MED 2011 event recognition dataset. For this set of experiments, we do not use the human detector during homography estimation. We took this decision for practical reasons: running the human detector on 1000 h of video would have taken more than two weeks on 500 cores; the speed is about 10–15 s/frame on a single core. We also leave out the T2 split of STP, because of both performance and computational reasons. We have found on a subset of TRECVID 2011 train data that the T2 of STP does not improve the results. This happens because the events do not have a temporal structure that can be easily captured by the rigid STP, as opposed to the actions that are temporally well cropped.

7.3.1 Evaluation of Improved Trajectory Features

Table 8 shows results on the TRECVID MED 2011 dataset. We contrast the different descriptors and their combinations for all the ten event categories. We observe that the MBH descriptors are best performing among the individual channels. The fact that HOG outperforms HOF demonstrates that there is rich contextual appearance information in the scene as TRECVID MED contains complex event videos.

Between the two-channel combinations, the best one is HOG + MBH, followed by HOG + HOF and HOF + MBH. This order is given by the complementarity of the features: both HOF and MBH encode motion information, while HOG captures texture information. Combining all three channels performs similarly to the best two-channel variant.

If we remove all spatio–temporal information (H3 and SFV), performance drops from 45.9 to 43.8. This underlines the importance of weak geometric information, even for the highly unstructured videos found in TRECVID MED.

We consider the effect of re-scaling the videos to different resolutions in Table 9 for both baseline DTF and our ITF. From the results we see that ITF always improves over DTF: even on low resolutions there are enough feature matches in order to estimate the homography reliably. The performance of both DTF and ITF does not improve much when using higher resolutions than 320.

The results in Table 9 also show that the gain from ITF on TRECVID MED is less pronounced than the gain observed for action recognition. This is possibly due to the generally poorer quality of the videos in this dataset, eg due to motion

Table 8 Performance in terms of AP on the full TRECVID MED 2011 dataset

	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 (%)
HOG	28.7	45.9	57.2	38.6	18.5	21.1	41.4	51.5	41.1	25.8	37.0
HOF	18.8	28.5	54.6	37.2	24.5	17.2	44.9	66.7	35.6	28.5	35.7
MBH	26.2	39.1	59.8	37.7	30.4	19.7	46.4	72.6	33.6	32.8	39.8
HOG + HOF	27.6	49.9	59.8	45.1	30.6	22.4	48.4	69.4	40.8	35.0	42.9
HOG + MBH	30.8	**53.9**	61.5	40.0	38.2	**28.8**	**53.4**	72.0	38.1	**43.3**	**46.0**
HOF + MBH	26.8	40.7	59.8	41.2	31.2	20.3	47.6	71.8	33.5	34.7	40.8
HOG + HOF +MBH	**31.3**	53.0	**61.9**	**47.4**	**38.2**	23.4	51.4	**73.2**	**41.6**	37.5	45.9

We use ITF and encode them with FV ($K = 256$). We also use SFV and STP, but only with a horizontal stride (H3), and no temporal split (T2). We rescale the video to a maximal width of 480 pixels
The best results for each class are shown in bold

Table 9 Comparison of our improved trajectory features (ITF) with the baseline dense trajectory features (DTF) for different resolutions on the TRECVID MED dataset

AP	160 px (%)	320 px (%)	480 px (%)	640 px (%)
DTF	40.6	44.9	43.0	44.3
ITF	41.0	45.6	**45.9**	45.4

For both ITF and DTF, we combine HOG, HOF and MBH, and use FV ($K = 256$) augmented with SFV and STP, but only use H3 and not T2 for STP
The best result is shown in bold

Table 10 The speed (frames/s) of computing our proposed video representation using different resolutions on the TRECVID MED dataset; left: the speed of computing raw features (i.e., DTF or ITF); right: the speed of encoding the features into a high dimensional Fisher vector ($K = 256$)

FPS	160 px		320 px		480 px		640 px	
DTF	40.8	83.4	10.4	22.1	4.5	9.2	2.1	5.2
ITF	18.5	91.7	5.1	23.8	2.2	10.2	1.2	5.9

blur in videos recorded by hand-held cameras. In addition, a major challenge in this data set is that for many videos the information characteristic for the category is limited to a relatively short sub-sequence of the video. As a result the video representations are affected by background clutter from irrelevant portions of the video. This difficulty might limit the beneficial effects of our improved features.

Table 10 provides the speed of computing our video representations when using the settings from Table 9. Computing ITF instead of DTF features increases the runtime by around of a factor of two. For our final setting (videos resized to 480 px width, improved dense trajectories, HOG, HOF, MBH, stabilized without the human detector and encoded with FV and H3 SPM and SFV), the slowdown factor with respect to the real video time is around 10× on a single core. This translates in less than a day of computation for the 1000 h of TRECVID test data on a 500-core cluster.

7.3.2 Comparison to the State of the Art

We compare to the state-of-the-art in Table 11. We consider the EVENTS/DEV-O split of the TRECVID MED 2011 dataset, since most results are reported using this setup.

The top three results were reported by the following authors. Li et al. (2013) attained 12.3 % by automatically segmenting videos into coherent sub-sequences over which the features are pooled. Vahdat et al. (2013) achieved 15.7 % by using multiple kernel learning to combine different features, and latent variables to infer the relevant portions of the videos. Tang et al. (2013) obtained the best reported result so far of 21.8 %, using a method based on AND-OR graphs to combine a large set of features in different subsets.

Table 11 Performance in terms of AP on the TRECVID MED 2011 dataset using the EVENTS/DEV-O split

Paper	Features	mAP
Tang et al. (2012)	HOG3D	4.8%
Vahdat and Mori (2013)	HOG3D, textual information	8.4%
Kim et al. (2013)	HOG3D, MFCC	9.7%
Li et al. (2013)	STIP	12.3%
Vahdat et al. (2013)	HOG3D, SSIM, color, sparse and dense SIFT	15.7%
Tang et al. (2013)	HOG3D, ISA, GIST, HOG, SIFT, LBP, texture, color	21.8%
ITF	HOG, HOF, MBH	**31.6**%

The feature settings are the same as Table 8: improved trajectory features (HOG + HOF + MBH), encoded with FV ($K = 256$) and SFV + H3
The bold value indicates the best performing method

We observe a dramatic improvement when comparing our result of 31.6 % to the state of the art. In contrast to these other approaches, our work focuses on good local features and their encoding, and then learns a linear SVM classifier over concatenated Fisher vectors computed from the HOG, HOF and MBH descriptors.

8 Conclusions

This paper improves dense trajectories by explicitly estimating camera motion. We show that the performance can be significantly improved by removing background trajectories and warping optical flow with a robustly estimated homography approximating the camera motion. Using a state-of-the-art human detector, possible inconsistent matches can be removed during camera motion estimation, which makes it more robust. We also explore Fisher vector as an alternative feature encoding approach to BOW histograms, and consider the effect of STP and SFVs to encode weak geometric layouts.

An extensive evaluation on three challenging tasks—action recognition, action localization in movies, and complex event recognition—demonstrates the effectiveness and flexibility of our new framework. We also found that action localization results can be substantially improved by using a simple re-scoring technique before applying NMS, to suppress a bias for too short windows. Our proposed pipeline significantly outperform the state of the art on all three tasks. Our approach can serve as a general pipeline for various video recognition problems.

Acknowledgments 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

Arandjelovic, R., & Zisserman, A. (2012). Three things everyone should know to improve object retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2911–2918).

Ballas, N., Yang, Y., Lan, Z. Z., Delezoide. B., Prêteux. F., & Hauptmann, A. (2013). Space-time robust video representation for action recognition. In *IEEE International Conference on Computer Vision*.

Bay, H., Tuytelaars, T., & Gool, L. V. (2006). SURF: Speeded up robust features. In *European Conference on Computer Vision*.

Cao, L., Mu, Y., Natsev, A., Chang, S. F., Hua, G., & Smith, J. (2012). Scene aligned pooling for complex video recognition. In *European Conference on Computer Vision*.

Chatfield, K., Lempitsky, V., Vedaldi, A., & Zisserman, A. (2011). The devil is in the details: An evaluation of recent feature encoding methods. In *British Machine Vision Conference*.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Dalal, N., Triggs, B., & Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *European Conference on Computer Vision*.

Dollár, P., Rabaud, V., Cottrell, G., & Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. In *IEEE Workshop Visual Surveillance and Performance Evaluation of Tracking and Surveillance*.

Duchenne, O., Laptev, I., Sivic, J., Bach, F., & Ponce, J. (2009). Automatic annotation of human actions in video. In *IEEE International Conference on Computer Vision* (pp. 1491–1498).

Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. In *Proceedings of the Scandinavian Conference on Image Analysis*.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(9), 1627–1645.

Ferrari, V., Marin-Jimenez, M., & Zisserman, A. (2008). Progressive search space reduction for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381–395.

Gaidon, A., Harchaoui, Z., & Schmid, C. (2011). Actom sequence models for efficient action detection. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Gaidon, A., Harchaoui, Z., & Schmid, C. (2013). Activity representation with motion hierarchies. *International Journal of Computer Vision*, *3*, 1–20.

Gauglitz, S., Höllerer, T., & Turk, M. (2011). Evaluation of interest point detectors and feature descriptors for visual tracking. *International Journal of Computer Vision*, *94*(3), 335–360.

van Gemert, J., Veenman, C., Smeulders, A., & Geusebroek, J. M. (2010). Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(7), 1271–1283.

Gorelick, L., Blank, M., Shechtman, E., Irani, M., & Basri, R. (2007). Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(12), 2247–2253.

Gupta, A., Kembhavi, A., & Davis, L. (2009). Observing human-object interactions: using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(10), 1775–1789.

Ikizler-Cinbis, N., & Sclaroff, S. (2010). Object, scene and actions: Combining multiple features for human action recognition. In *European Conference on Computer Vision*.

Izadinia, H., & Shah, M. (2012). Recognizing complex events using large margin joint low-level event model. In *European Conference on Computer Vision*.

Jain, M., Jégou, H., & Bouthemy, P. (2013). Better exploiting motion for better action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., & Schmid, C. (2011). Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Jiang, Y. G., Dai, Q., Xue, X., Liu, W., & Ngo, C. W. (2012). Trajectory-based modeling of human actions with motion reference points. In *European Conference on Computer Vision* (pp. 425–438).

Jiang, Y. G., Liu, J., Roshan Zamir, A., Laptev, I., Piccardi, M., Shah, M., & Sukthankar, R. (2013). THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/ICCV13-Action-Workshop/

Karaman, S., Seidenari, L., Bagdanov, A. D., & Del Bimbo, A. (2013). L1-regularized logistic regression stacking and transductive crf smoothing for action recognition in video. In *ICCV Workshop on Action Recognition with a Large Number of Classes*.

Kim, I., Oh, S., Vahdat, A., Cannons, K., Perera, A., & Mori, G. (2013). Segmental multi-way local pooling for video recognition. In *ACM Conference on Multimedia* (pp. 637–640).

Kläser, A., Marszałek, M., & Schmid, C. (2008). A spatio-temporal descriptor based on 3D-gradients. In *British Machine Vision Conference*

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

Krapac, J., Verbeek, J., & Jurie, F. (2011). Modeling spatial layout with Fisher vectors for image categorization. In *IEEE International Conference on Computer Vision*.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., & Serre, T. (2011). HMDB: A large video database for human motion recognition. In *IEEE International Conference on Computer Vision* (pp. 2556–2563).

Laptev, I. (2005). On space-time interest points. *International Journal of Computer Vision*, *64*(2–3), 107–123.

Laptev, I., & Pérez, P. (2007). Retrieving actions in movies. In *IEEE International Conference on Computer Vision*.

Laptev, I., Marszałek, M., Schmid, C., & Rozenfeld, B. (2008). Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Le, Q. V., Zou, W. Y., Yeung, S. Y., & Ng, A. Y. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Li, K., Oh, S., Perera, A. A., & Fu, Y. (2012). A videography analysis framework for video retrieval and summarization. In *British Machine Vision Conference* (pp. 1–12).

Li, W., Yu, Q., Divakaran, A., & Vasconcelos, N. (2013). Dynamic pooling for complex event recognition. In *IEEE International Conference on Computer Vision*.

Liu, J., Luo, J., & Shah, M. (2009). Recognizing realistic actions from videos in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Ma, S., Zhang, J., Ikizler-Cinbis, N., & Sclaroff, S. (2013). Action recognition and localization by hierarchical space-time segments. In *IEEE International Conference on Computer Vision*.

Marszałek, M., Laptev, I., & Schmid, C. (2009). Actions in context. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Mathe, S., & Sminchisescu, C. (2012). Dynamic eye movement datasets and learnt saliency models for visual action recognition. In *European Conference on Computer Vision* (pp. 842–856).

Matikainen, P., Hebert, M., & Sukthankar, R. (2009). Trajectons: Action recognition through the motion analysis of tracked features. In *ICCV Workshops on Video-Oriented Object and Event Classification*.

Matikainen, P., Hebert, M., & Sukthankar, R. (2010). Representing pairwise spatial and temporal relations for action recognition. In *European Conference on Computer Vision*.

McCann, S., & Lowe, D. G. (2013). Spatially local coding for object recognition. In *Asian Conference on Computer Vision* (pp. 204–217). New York: Springer.

Messing, R., Pal, C., & Kautz, H. (2009). Activity recognition using the velocity histories of tracked keypoints. In *IEEE International Conference on Computer Vision*.

Murthy, O. R., & Goecke, R. (2013a). Combined ordered and improved trajectories for large scale human action recognition. In *ICCV Workshop on Action Recognition with a Large Number of Classes*.

Murthy, O. R., & Goecke, R. (2013b). Ordered trajectories for large scale human action recognition. In *ICCV Workshops*.

Natarajan, P., Wu, S., Vitaladevuni, S., Zhuang, X., Tsakalidis, S., Park, U., Prasad, R., & Natarajan, P. (2012). Multimodal feature fusion for robust event detection in web videos. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Niebles, J. C., Chen, C. W., & Fei-Fei, L. (2010). Modeling temporal structure of decomposable motion segments for activity classification. In *European Conference on Computer Vision*

Oneata, D., Verbeek, J., & Schmid, C. (2013). Action and event recognition with Fisher vectors on a compact feature set. In *IEEE International Conference on Computer Vision*.

Over, P., Awad, G., Michel, M., Fiscus, J., Sanders, G., Shaw, B., Kraaij, W., Smeaton, A. F., & Quenot, G. (2012). TRECVID 2012: An overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID*.

Park, D., Zitnick, C. L., Ramanan, D., & Dollár, P. (2013). Exploring weak stabilization for motion feature extraction. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Patron-Perez, A., Marszalek, M., Zisserman, A., & Reid, I. (2010). High Five: Recognising human interactions in TV shows. In *British Machine Vision Conference*.

Peng, X., Wang, L., Cai, Z., Qiao, Y., & Peng, Q. (2013). Hybrid super vector with improved dense trajectories for action recognition. In *ICCV Workshops*

Prest, A., Schmid, C., & Ferrari, V. (2012). Weakly supervised learning of interactions between humans and objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(3), 601–614.

Prest, A., Ferrari, V., & Schmid, C. (2013). Explicit modeling of human-object interactions in realistic videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(4), 835–848.

Reddy, K., & Shah, M. (2012). Recognizing 50 human action categories of web videos. *Machine Vision and Applications* (pp. 1–11).

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

Sánchez, J., Perronnin, F., Mensink, T., & Verbeek, J. (2013). Image classification with the Fisher vector: Theory and practice. *International Journal of Computer Vision*, *105*(3), 222–245.

Sapienza, M., Cuzzolin, F., & Torr, P. (2012). Learning discriminative space-time actions from weakly labelled videos. In *British Machine Vision Conference*.

Schüldt, C., Laptev, I., & Caputo, B. (2004). Recognizing human actions: A local SVM approach. In *International Conference on Pattern Recognition*.

Scovanner, P., Ali, S., & Shah, M. (2007). A 3-dimensional SIFT descriptor and its application to action recognition. In *ACM Conference on Multimedia*.

Shi, F., Petriu, E., & Laganiere, R. (2013). Sampling strategies for real-time action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Shi, J., & Tomasi, C. (1994). Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Soomro, K., Zamir, A. R., & Shah, M. (2012). UCF101: A dataset of 101 human actions classes from videos in the wild. CRCV-TR-12-01.

Sun, C., & Nevatia, R. (2013). Large-scale web video event classification by use of fisher vectors. In *IEEE Winter Conference on Applications of Computer Vision*.

Sun, J., Wu, X., Yan, S., Cheong, L. F., Chua, T. S., & Li, J. (2009). Hierarchical spatio-temporal context modeling for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Szeliski, R. (2006). Image alignment and stitching: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, *2*(1), 1–104.

Tang, K., Fei-Fei, L., & Koller, D. (2012). Learning latent temporal structure for complex event detection. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1250–1257).

Tang, K., Yao, B., Fei-Fei, L., & Koller, D. (2013). Combining the right features for complex event recognition. In *IEEE International Conference on Computer Vision* (pp. 2696–2703).

Uemura, H., Ishikawa, S., & Mikolajczyk, K. (2008). Feature tracking and motion compensation for action recognition. In *British Machine Vision Conference*.

Vahdat, A., & Mori, G. (2013). Handling uncertain tags in visual recognition. In *IEEE International Conference on Computer Vision*.

Vahdat, A., Cannons, K., Mori, G., Oh, S., & Kim, I. (2013). Compositional models for video event detection: A multiple kernel learning latent variable approach. In *IEEE International Conference on Computer Vision*.

Vig, E., Dorr, M., & Cox, D. (2012). Space-variant descriptor sampling for action recognition based on saliency and eye movements. In *European Conference on Computer Vision* (pp. 84–97).

Wang, H., & Schmid, C. (2013). Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*.

Wang, H., Ullah, M. M., Kläser, A., Laptev, I., & Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference*.

Wang, H., Kläser, A., Schmid, C., & Liu, C. L. (2013a). Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, *103*(1), 60–79.

Wang, L., Qiao, Y., Tang, X., et al (2013b). Mining motion atoms and phrases for complex action recognition. In *IEEE International Conference on Computer Vision* (pp. 2680–2687).

Wang, X., Wang, L., & Qiao, Y. (2012). A comparative study of encoding, pooling and normalization methods for action recognition. In *Asian Conference on Computer Vision*.

Willems, G., Tuytelaars, T., & Gool, L. (2008). An efficient dense and scale-invariant spatio-temporal interest point detector. In *European Conference on Computer Vision*.

Wu, S., Oreifej, O., & Shah, M. (2011). Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories. In *IEEE International Conference on Computer Vision*.

Yang, Y., & Shah, M. (2012). Complex events detection using data-driven concepts. In *European Conference on Computer Vision*.

Yeffet, L., & Wolf, L. (2009). Local trinary patterns for human action recognition. In *IEEE International Conference on Computer Vision*.

Yu, G., Yuan, J., & Liu, Z. (2012). Propagative Hough voting for human activity recognition. In *European Conference on Computer Vision* (pp. 693–706).

Zhu, J., Wang, B., Yang, X., Zhang, W., & Tu, Z. (2013). Action recognition with actons. In *IEEE International Conference on Computer Vision*.