Action Recognition with Improved Trajectories

Heng Wang, Cordelia Schmid

Action Recognition with Improved Trajectories

Heng Wang and Cordelia Schmid
LEAR, INRIA, France
firstname.lastname@inria.fr

Abstract

Recently dense trajectories were shown to be an efficient video representation for action recognition and achieved state-of-the-art results on a variety of datasets. This paper improves their performance by taking into account camera motion to correct them. To estimate camera motion, we match feature points between frames using SURF descriptors and dense optical flow, which are shown to be complementary. These matches are, then, used to robustly estimate a homography with RANSAC. Human motion is in general different from camera motion and generates inconsistent matches. To improve the estimation, a human detector is employed to remove these matches. Given the estimated camera motion, we remove trajectories consistent with it. We also use this estimation to cancel out camera motion from the optical flow. This significantly improves motion-based descriptors, such as HOF and MBH. Experimental results on four challenging action datasets (i.e., Hollywood2, HMDB51, Olympic Sports and UCF50) significantly outperform the current state of the art.

1. Introduction

Action recognition has been an active research area for over three decades. Recent research focuses on realistic datasets collected from movies [20, 22], web videos [21, 31], TV shows [28], *etc*. These datasets impose significant challenges on action recognition, *e.g*., background clutter, fast irregular motion, occlusion, viewpoint changes. Local space-time features [7, 19] were shown to be successful on these datasets, since they avoid non-trivial pre-processing steps, such as tracking or segmentation. A bag-of-features representation of these local features can be directly used for action classification and achieves state-of-the-art performance (see [1] for a recent survey).

Many classical image features have been generalized to videos, *e.g*., 3D-SIFT [33], extended SURF [41], HOG3D [16], and local trinary patterns [43]. Among the local space-time features, dense trajectories [40] have been shown to perform 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) [6] give the best results due to their robustness to camera motion.

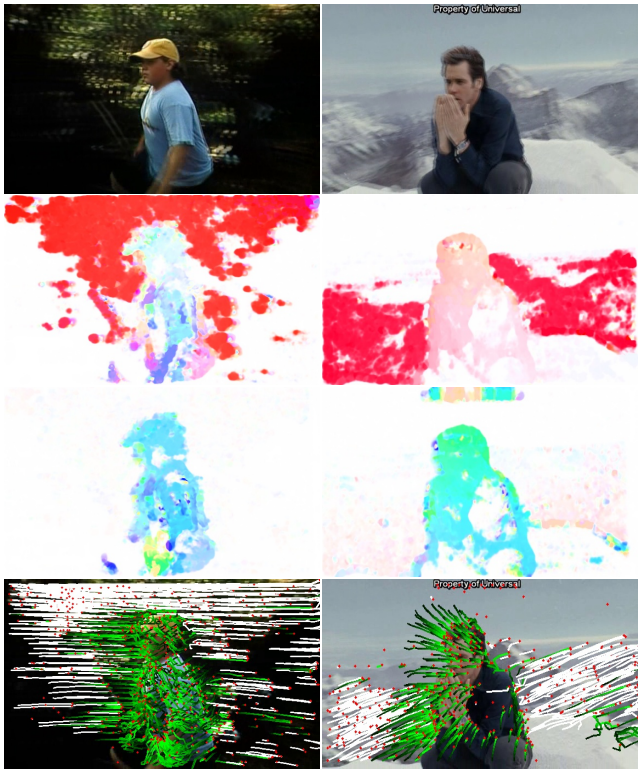

Figure 1. First row: images of two consecutive frames overlaid; second row: optical flow [8] between the two frames; third row: optical flow after removing camera motion; last row: trajectories removed due to camera motion in white.

MBH is based on derivatives of optical flow, which is a simple and efficient way to suppress camera motion. However, we argue that we can still benefit from explicit camera motion estimation. Camera motion generates many irrele-

Figure 2. Visualization of inlier matches of the robustly estimated homography. Green arrows correspond to SURF descriptor matches, and red ones to dense optical flow.

vant trajectories in the background in realistic videos. We can prune them and only keep trajectories from humans or objects of interest, if we know the camera motion (see Figure 1). Furthermore, given the camera motion, we can correct the optical flow, so that the motion vectors of human actors are independent of camera motion. This improves the performance of motion descriptors based on optical flow, *i.e*., HOF (histograms of optical flow) and MBH. We illustrate the difference between the original and corrected optical flow in the middle two rows of Figure 1.

Very few approaches consider camera motion when extracting feature trajectories for action recognition. Uemura *et al*. [38] combine feature matching with image segmentation to estimate the dominant camera motion, and then separate feature tracks from the background. Wu *et al*. [42] apply a low-rank assumption to decompose feature trajectories into camera-induced and object-induced components. Recently, Park *et al*. [27] perform weak stabilization to remove both camera and object-centric motion using coarse-scale optical flow for pedestrian detection and pose estimation in video. Jain *et al*. [14] decompose visual motion into dominant and residual motions both for extracting trajectories and computing descriptors.

Among the approaches improving dense trajectories, Vig *et al*. [39] 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*. [15] cluster dense trajectories, and use the cluster centers as reference points so that the relationship between them can be modeled.

The rest of the paper is organized as follows. In section 2, we detail our approach for camera motion estimation and discuss how to remove inconsistent matches due to humans. Experimental setup and evaluation protocols are explained in section 3 and experimental results in section 4. The code to compute improved trajectories and descriptors is available online.[1]

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

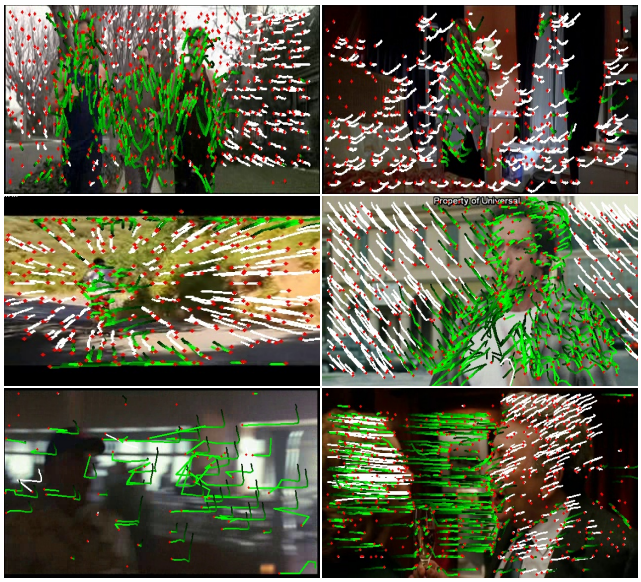

Figure 3. Examples of removed trajectories under various camera motions, *e.g*., pan, zoom, tilt. White trajectories are considered due to camera motion. The red dots are the trajectory positions in the current frame. The last row shows two failure cases. The left one is due to severe motion blur. The right one fits the homography to the moving humans as they dominate the frame.

2. Improving dense trajectories

In this section, we first describe the major steps of our camera motion estimation method, and how to use it to improve dense trajectories. We, then, discuss how to remove potentially inconsistent matches based on humans to obtain a robust homography estimation.

2.1. Camera motion estimation

To estimate the global background motion, we assume that two consecutive frames are related by a homography [37]. 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 SURF [3] features and match them based on the nearest neighbor rule. The reason for choosing SURF features is their robustness to motion blur, as shown in a recent evaluation [13].

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 [8]. We select motion vectors for salient feature points using the good-features-to-track criterion [35], *i.e*., thresholding the smallest eigenvalue of the autocorrelation matrix.

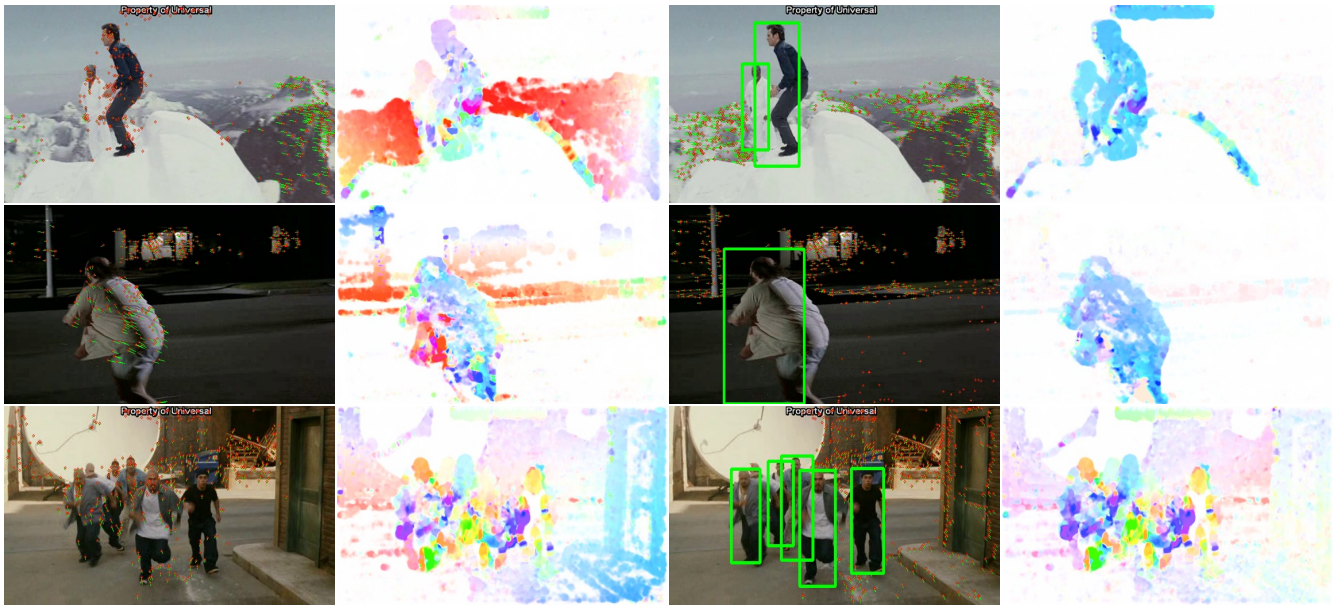

Figure 4. Homography estimation without human detector (left) and with human detector (right). We show inlier matches in the first and third columns. The optical flow (second and fourth columns) is warped with the corresponding homography. The first and second rows show a clear improvement of the estimated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

The two approaches are complementary. SURF focuses on blob-type structures, whereas [35] fires on corners and edges. Figure 2 visualizes the two types of matches in different colors. Combining them results in a more balanced distribution of matched points, which is critical for a good homography estimation.

We, then, robustly estimate the homography using RANSAC [11]. This allows us to rectify the image to remove the camera motion. Figure 1 (two rows in the middle) demonstrates the difference of optical flow before and after rectification. Compared to the original flow (the second row of Figure 1), the rectified version (the third row) suppresses the background camera motion and enhances the foreground moving objects.

For dense trajectories, 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 [40], the performance of the HOF descriptor degrades significantly in the presence of camera motion. Our experimental results (in section 4.1) show that HOF can achieve similar performance as MBH when we have correct foreground 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 too 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 (*e.g.*, pan, tilt and zoom) and only trajectories related to human actions are kept (shown in green in Figure 3). This gives us similar effects as sampling features based on visual saliency maps [23, 39].

The last row of Figure 3 shows two failure cases. The left 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 example shown on the right, humans dominate the frame, which causes homography estimation to fail. We discuss a solution for such cases in the following section.

2.2. 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 there are dramatic pose changes when the person is performing the action. Furthermore, the person could only be visible partially due to occlusion or being partially out of view.

Here, we apply a state-of-the-art human detector [30], which adapts the general part-based human detector [9] 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 [10]. 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 Figure 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 Figure 4), when the human bounding boxes are used to remove matches not corresponding to camera motion. The last row of Figure 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 section 4.3, we compare the performance of action recognition with or without human detection.

The human detector does not always work perfectly. It can miss humans due to pose or viewpoint changes. 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, *i.e*., we take the average flow vector [8] 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.[1] In the following, we always use the human detector to remove potentially inconsistent matches before computing the homography, unless stated otherwise.

3. Experimental setup

In this section, we first present implementation details for our trajectory features. We, then, introduce the feature encoding used in our evaluation. Finally, the datasets and experimental setup are presented.

3.1. Trajectory features

We, first, briefly describe the dense trajectory features [40], which are used as the baseline in our experiments. The approach densely samples 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 [8]. 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 several descriptors (*i.e*., Trajectory, HOG, HOF and MBH) with exactly the same parameters as [40]. The Trajectory descriptor is a concatenation of normalized displacement vectors. The other descriptors are computed in the space-time volume aligned with the trajectory. HOG is based on the orientation of image gradients and captures the static appearance information. Both HOF and MBH 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 30 for Trajectory, 96 for HOG, 108 for HOF and 192 for MBH.

To normalize the histogram-based descriptors, *i.e*., HOG, HOF and MBH, we apply the recent RootSIFT [2] approach, *i.e*., square root each dimension after L1 normalization. We do not perform L2 normalization as in [40]. This brings about 0.5% improvement for the histogram-based descriptors. We use this normalization in all the experiments.

To extract our improved trajectories, we sample and track feature points exactly the same way as [40], see above. To compute the descriptors, we first estimate the homography with RANSAC using the feature matches extracted between two consecutive frames; matches on detected humans are removed. We, then, warp the second frame with the estimated homography. The optical flow [8] is 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 in the baseline.

The Trajectory descriptor is also computed based on the motion vectors of the warped flow. We further utilize these stabilized motion vectors to remove background trajectories. For each trajectory, we compute the maximal magnitude of them. If the maximal magnitude is lower than a threshold (*i.e*., 1 pixel), the trajectory is considered to be consistent with camera motion, and thus removed.

3.2. Feature encoding

To encode features, we use bag of features and Fisher vector. For bag of features, we use identical settings to [40]. We train a codebook for each descriptor type using 100,000 randomly sampled features with k-means. The size of the codebook is set to 4000. An SVM with RBF-χ^2 kernel is used for classification, and different descriptor types are

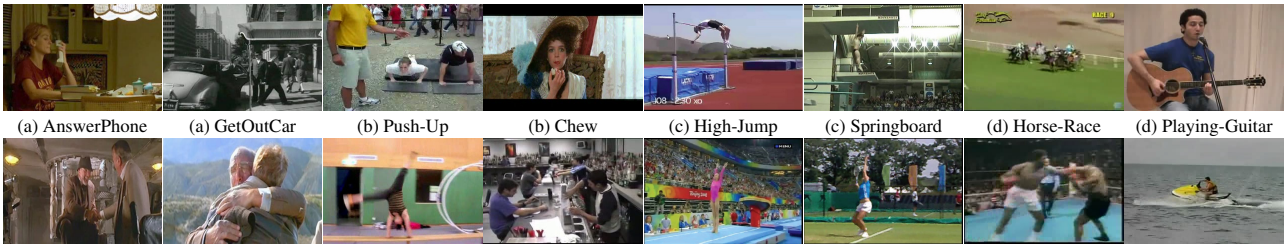

Figure 5. From left to right, example frames from (a) Hollywood2, (b) HMDB51, (c) Olympic Sports and (d) UCF50.

combined by summing their kernel matrices normalized by the average distance.

Unlike bag of features, Fisher vector [29] encodes both first and second order statistics between the video descriptors and a Gaussian Mixture Model (GMM). In recent evaluations [5, 26], this shows an improved performance over bag of features for both image and action classification. Differently from the bag-of-features encoding, we first reduce the descriptor dimensionality by a factor of two using Principal Component Analysis (PCA), as in [29]. We set the number of Gaussians to $K = 256$ and randomly sample a subset of 256,000 features from the training set to estimate the GMM. Each video is, then, represented by a $2DK$ dimensional Fisher vector for each descriptor type, where D is the descriptor dimension after performing PCA. Finally, we apply power and L2 normalization to the Fisher vector, as in [29]. To combine different descriptor types, we concatenate their normalized Fisher vectors. A linear SVM is used for classification.

In all experiments we fix $C = 100$ for the SVM, which has shown to give good results when validating on a subset of training samples. In the case of multi-class classification, we use a one-against-rest approach and select the class with the highest score. In the following, we use Fisher vector encoding unless stated otherwise, since it results in better performance, see section 4.2.

3.3. Datasets

This section briefly describes the four datasets (Hollywood2, HMDB51, Olympic Sports and UCF50) used in our experiments, see Figure 5. These are among the most challenging datasets in the literature.

The **Hollywood2** dataset [22] has been collected from 69 different Hollywood movies and includes 12 action classes. It contains 1,707 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 [22].

The **HMDB51** dataset [18] is collected from a variety of sources ranging from digitized movies to YouTube videos. In total, there are 51 action categories and 6,766 video sequences. We follow the original protocol using three train-test splits [18]. 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 our experiments we use the original videos and not the stabilized ones.

The **Olympic Sports** dataset [24] 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 [24].

The **UCF50** dataset [31] 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 4 action clips. In total, there are 6,618 video clips. The video clips in the same group may share some common features, such as the same person, similar background or similar viewpoint. We apply the leave-one-group-out cross-validation as recommended by the authors and report average accuracy over all classes.

4. Experimental results

We, first, evaluate the gain due to different motion stabilization steps in section 4.1. Section 4.2 measures the improvement for bag of features and Fisher vector and compares the two. Section 4.3 evaluates the impact of removing inconsistent matches based on human detection. Finally, we compare with the state of the art in section 4.4.

4.1. Evaluation of improved dense trajectories

We choose the dense trajectories [40] as our baseline and apply RootSIFT normalization as described in section 3.1. 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 corresponding to the camera motion and removing background trajectories consistent with the homography. The combined setting uses both. The results are presented in Table 1.

	Hollywood2				HMDB51			
	Baseline	WarpFlow	RmTrack	Combined	Baseline	WarpFlow	RmTrack	Combined
Trajectory	42.2%	47.6%	42.4%	48.5%	25.4%	31.0%	26.9%	32.4%
HOG	46.9%	46.2%	46.7%	47.1%	38.4%	38.7%	39.6%	40.2%
HOF	51.4%	58.1%	53.4%	58.8%	39.5%	48.5%	41.6%	48.9%
MBH	57.4%	60.3%	58.6%	60.5%	49.1%	50.9%	50.8%	52.1%
HOF+MBH	58.2%	62.3%	59.7%	62.6%	49.8%	53.5%	51.0%	54.7%
Combined	60.1%	63.6%	61.7%	64.3%	52.2%	55.6%	53.9%	57.2%
	Olympic Sports				UCF50			
	Baseline	WarpFlow	RmTrack	Combined	Baseline	WarpFlow	RmTrack	Combined
Trajectory	62.4%	73.7%	66.3%	77.2%	65.3%	72.6%	67.8%	75.2%
HOG	77.0%	76.3%	78.7%	78.8%	81.8%	81.6%	82.6%	82.6%
HOF	74.5%	86.2%	77.6%	87.6%	74.3%	85.4%	79.4%	85.1%
MBH	82.4%	87.5%	86.0%	89.1%	86.5%	88.4%	88.0%	88.9%
HOF+MBH	82.1%	88.3%	86.2%	89.7%	87.1%	89.3%	87.5%	89.5%
Combined	84.7%	88.9%	87.0%	91.1%	88.6%	90.9%	88.9%	91.2%

Table 1. Comparison of the baseline with our method and two intermediate results using FV encoding. "WarpFlow": computing motion descriptors (*i.e.*, Trajectory, HOF and MBH) using warped optical flow, while keep all the trajectories; "RmTrack": removing background trajectories, but computing motion descriptors using the original flow field; "Combined": removing background trajectories, and computing Trajectory, HOF and MBH with warped optical flow.

In the following, we discuss the results descriptor by descriptor. The performance of the Trajectory descriptor is significantly improved, when camera motion is compensated for. On Olympic Sports, there is over 10% improvement w.r.t. the baseline method. On the other three datasets, we also have over 5% improvement. "Combined" further improves over "WarpFlow" as background trajectories are removed.

The results of HOG are very similar for different variants on all four datasets. Since HOG is designed to capture static appearance information, we do not expect that compensating camera motion significantly improves its performance. We observe small improvements of around 1%, which is probably due to removing background trajectories.

HOF benefits most from stabilizing optical flow. Both "Combined" and "WarpFlow" are significantly better than the other two. On all datasets, the improvements are over 5%. On HMDB51, Olympic Sports, and UCF50, the improvements are even higher, *i.e.*, around 10%. After motion compensation, the performance of HOF is now comparable to MBH.

MBH is known for its robustness to camera motion [40]. However, its performance still improves, as motion boundaries are much clearer, see Figures 1 and 4. We have over 3% improvement on Hollywood2, HMDB51 and Olympic Sports for MBH.

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. On Hollywood2 and HMDB51, "HOF+MBH" is over 2% better than MBH or HOF alone. Combining all the descriptors further increases the performance, as shown in the last row of each dataset.

4.2. Feature encoding with BOF and FV

In this section, we evaluate the performance of our improved trajectories using different feature encoding methods. Table 2 compares the final performance of all four descriptors combined. We can observe a similar amount of improvement due to our motion stabilized descriptors when encoding them with bag of features (BOF) or Fisher vector (FV). As our approach focuses on the local descriptor level, its improvement is independent of the feature encoding method. For example, "ITF" is around 4% (5%) better than "DTF" on Hollywood2 (HMDB51) for both bag of features and Fisher vector. Furthermore, a Fisher vector representation always results in a better performance than bag of features for both "DTF" and "ITF". The improvement varies from dataset to dataset. On Hollywood2 it is around 2%, whereas on Olympic Sports it is over 7%. Note

Datasets	Bag of features		Fisher vector	
	DTF	ITF	DTF	ITF
Hollywood2	58.5%	62.2%	60.1%	64.3%
HMDB51	47.2%	52.1%	52.2%	57.2%
Olympic Sports	75.4%	83.3%	84.7%	91.1%
UCF50	84.8%	87.2%	88.6%	91.2%

Table 2. Comparison of feature encoding with bag of features and Fisher vector. "DTF" stands for the original dense trajectory features [40] with RootSIFT normalization, whereas "ITF" are our improved trajectory features.

Hollywood2		HMDB51		Olympic Sports		UCF50	
Vig *et al*. [39]	59.4%	Sadanand *et al*. [32]	26.9%	Brendel *et al*. [4]	77.3%	Kliper-Gross *et al*. [17]	72.7%
Jiang *et al*. [15]	59.5%	Kliper-Gross *et al*. [17]	29.2%	Jiang *et al*. [15]	80.6%	Solmaz *et al*. [36]	73.7%
Mathe *et al*. [23]	61.0%	Jiang *et al*. [15]	40.7%	Gaidon *et al*. [12]	82.7%	Reddy *et al*. [31]	76.9%
Jain *et al*. [14]	62.5%	Jain *et al*. [14]	52.1%	Jain *et al*. [14]	83.2%	Shi *et al*. [34]	83.3%
Without HD	63.0%	Without HD	55.9%	Without HD	90.2%	Without HD	90.5%
With HD	**64.3%**	With HD	**57.2%**	With HD	**91.1%**	With HD	**91.2%**

Table 4. Comparison of our results to the state of art. We present our results for FV encoding both with and without automatic human detection (HD).

that for bag of features we use an SVM with RBF-χ^2 kernel, whereas for Fisher vector we use a linear SVM, see section 3.2.

4.3. Removing inconsistent matches due to humans

In this section, we investigate the impact of removing inconsistent matches due to humans when estimating the homography, see Figure 4 for an illustration. We compare three cases, *i.e*., estimating the homography without human detection, with automatic human detection, and 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. To limit the labeling effort, we annotated humans in 20 training and 20 testing videos for each action class from Hollywood2.

As shown in Table 3, human detection helps to improve all motion related descriptors (Trajectory, HOF and MBH), since removing inconsistent matches on humans improves the homography estimation. Typically, the results are improved by around 2% when using an automatic human detector. If the humans are labeled by hand, we can further improve the performance by 1%.

The last two rows of Table 4 show the impact of automatic human detection on all four datasets. It is always better to use human detection for homography estimation on these action datasets. On Hollywood2 and HMDB51, the improvements are over 1%. Both datasets contain a huge amount of movies, where humans often occupy a large part of the image. On the other two datasets, the impact is less pronounced as humans occupy smaller areas in the image.

Hollywood2-sub	None	Automatic	Manual
Trajectory	32.3%	35.7%	37.1%
HOG	34.5%	34.9%	34.7%
HOF	43.9%	45.2%	46.7%
MBH	45.8%	47.4%	49.2%
Combined	48.9%	50.7%	51.9%

Table 3. Comparison of the results on a subset of the Hollywood2 dataset with FV encoding. "None": without human detection; "Automatic": automatic human detection; "Manual": manual labeling of humans.

4.4. Comparison to the state of the art

Table 4 compares our method with the most recent results reported in the literature for all four datasets. On Hollywood2, all presented results [14, 15, 23, 39] improve dense trajectories in different ways. Jiang *et al*. [15] model the relationship between dense trajectory clusters. Both Mathe *et al*. [23] and Vig *et al*. [39] prune background features based on visual saliency. Here we only compare to their results when the saliency map is extracted automatically. Recently, Jain *et al*. [14] report 62.5% by decomposing visual motion to stabilize dense trajectories. We further improve their results by around 2%.

HMDB51 [18] is a relatively new dataset. Recently, Sadanand and Corso [32] report 26.9% with a high-level semantic representation of actions. Kliper-Gross *et al*. [17] improve it to 29.2% using motion interchange patterns. Dense trajectories based approaches [14, 15] seem to be very successful on HMDB51. The previous best result is from [14]. We improve it further by around 5%, and obtain 57.2% accuracy.

Olympic Sports [24] is collected from sports videos. It contains significant camera motion, which results in a large number of trajectories in the background. This dataset is also known to have rich structure information, where graph models [4] are shown to work very well. Gaidon *et al*. [12] report 82.7% by modeling trajectory clusters with a tree structure. Jain *et al*. [14] achieve a slightly better performance of 83.2% with motion decomposition. We improve their results by around 8%.

UCF50 [31] is an extension of the YouTube dataset [21]. Solmaz *et al*. [36] report 73.7% with a GIST3D video descriptor, an extension of the GIST descriptor [25] to video. Reddy and Shah [31] achieve 76.9% by combining the MBH descriptor with scene context information. Recently, Shi *et al*. [34] report 83.3% using randomly sampled HOG, HOF, HOG3D and MBH descriptors. We significantly improve over their result by around 8%.

5. Conclusion

This paper improves dense trajectories by explicitly estimating camera motion. We show that the performance can be significantly improved by removing background trajecto-

ries and warping optical flow with a robustly estimated homography approximating the camera motion. Using a state-of-the-art human detector, potentially inconsistent matches can be removed during camera motion estimation, which makes it more robust. An extensive evaluation on four challenging datasets demonstrates the effectiveness of the proposed approach, and establishes new bounds of performance.

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

[1] J. Aggarwal and M. Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 43(3):16:1–16:43, 2011.

[2] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, pages 2911–2918, 2012.

[3] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. In *ECCV*, 2006.

[4] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *ICCV*, 2011.

[5] 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.

[6] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006.

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

[8] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *SCIA*, 2003.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE PAMI*, 32(9):1627–1645, 2010.

[10] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, 2008.

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

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

[13] S. Gauglitz, T. Höllerer, and M. Turk. Evaluation of interest point detectors and feature descriptors for visual tracking. *IJCV*, 94(3):335–360, 2011.

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

[15] 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.

[16] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *BMVC*, 2008.

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

[18] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *ICCV*, pages 2556–2563, 2011.

[19] I. Laptev. On space-time interest points. *IJCV*, 64(2-3):107–123, 2005.

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

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

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

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

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

[25] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[26] D. Oneata, J. Verbeek, and C. Schmid. Action and event recognition with Fisher vectors on a compact feature set. In *ICCV*, 2013.

[27] D. Park, C. L. Zitnick, D. Ramanan, and P. Dollár. Exploring weak stabilization for motion feature extraction. In *CVPR*, 2013.

[28] A. Patron-Perez, M. Marszalek, I. Reid, and A. Zisserman. Structured learning of human interactions in TV shows. *IEEE PAMI*, 2012.

[29] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *ECCV*, 2010.

[30] A. Prest, C. Schmid, and V. Ferrari. Weakly supervised learning of interactions between humans and objects. *IEEE PAMI*, 34(3):601–614, 2012.

[31] K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, pages 1–11, 2012.

[32] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012.

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

[34] F. Shi, E. Petriu, and R. Laganiere. Sampling strategies for real-time action recognition. In *CVPR*, 2013.

[35] J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994.

[36] B. Solmaz, S. M. Assari, and M. Shah. Classifying web videos using a global video descriptor. *Machine Vision and Applications*, pages 1–13, 2012.

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

[38] H. Uemura, S. Ishikawa, and K. Mikolajczyk. Feature tracking and motion compensation for action recognition. In *BMVC*, 2008.

[39] E. Vig, M. Dorr, and D. Cox. Space-variant descriptor sampling for action recognition based on saliency and eye movements. In *ECCV*, 2012.

[40] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 103(1):60–79, 2013.

[41] G. Willems, T. Tuytelaars, and L. Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, 2008.

[42] S. Wu, O. Oreifej, and M. Shah. Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories. In *ICCV*, 2011.

[43] L. Yeffet and L. Wolf. Local trinary patterns for human action recognition. In *ICCV*, 2009.