Dense trajectories and motion boundary descriptors for action recognition

Heng Wang, Alexander Kläser, Cordelia Schmid, Cheng-Lin Liu

Dense trajectories and motion boundary descriptors for action recognition

Heng Wang, Alexander Kläser, Cordelia Schmid, Cheng-Lin Liu

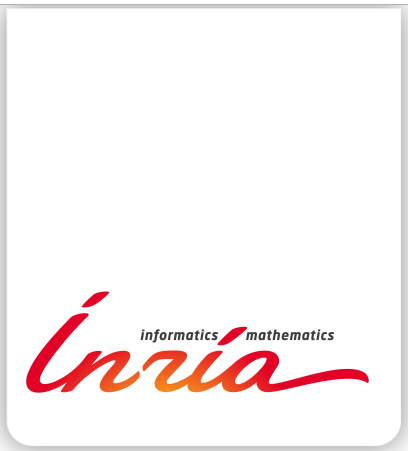
RESEARCH REPORT

N° 8050

August 2012

Project-Teams LEAR and NLPR CASIA

ISSN 0249-6399 ISRN INRIA/RR--8050--FR+ENG



Dense trajectories and motion boundary descriptors for action recognition

Heng Wang*, Alexander Kläser†, Cordelia Schmid†, Cheng-Lin Liu*

Project-Teams LEAR and NLPR CASIA

Research Report n° 8050 — August 2012 — 30 pages

Abstract: This paper introduces a video representation based on dense trajectories and motion boundary descriptors. Trajectories capture the local motion information of the video. A dense representation guarantees a good coverage of foreground motion as well as of the surrounding context. A state-of-the-art optical flow algorithm enables a robust and efficient extraction of dense trajectories.
As descriptors we extract features aligned with the trajectories to characterize shape (point coordinates), appearance (histograms of oriented gradients) and motion (histograms of optical flow). Additionally, we introduce a descriptor based on motion boundary histograms (MBH) which rely on differential optical flow. The MBH descriptor shows to consistently outperform other state-of-the-art descriptors, in particular on real-world videos that contain a significant amount of camera motion.
We evaluate our video representation in the context of action classification on nine datasets, namely KTH, YouTube, Hollywood2, UCF sports, IXMAS, UIUC, Olympic Sports, UCF50 and HMDB51. On all datasets our approach outperforms current state-of-the-art results.

Key-words: Action recognition, Dense trajectories, Motion boundary histograms

* National Laboratory of Pattern Recognition, CASIA
† LEAR team, INRIA Grenoble Rhône-Alpes

RESEARCH CENTRE GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Résumé :

Mots-clés :

Contents

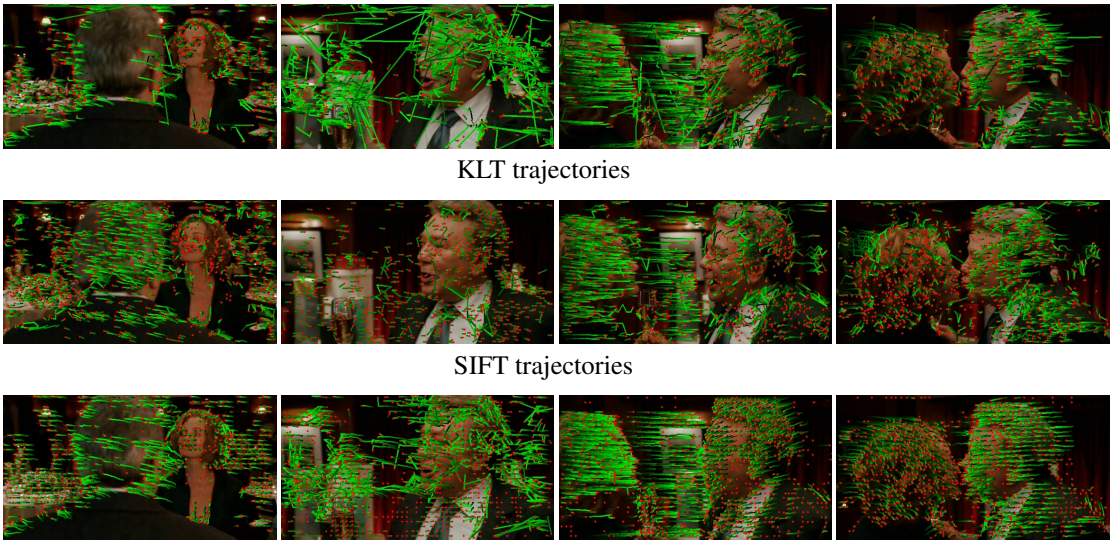


Figure 1: Visualization of KLT, SIFT and dense trajectories for a “kiss" action. Red dots indicate the point positions in the current frame. Compared to KLT trajectories, dense trajectories are more robust to fast irregular motions, in particular at shot boundaries (second column). SIFT trajectories can also handle shot boundaries, but are not able to capture the complex motion patterns accurately.

1 Introduction

Local space-time features are a successful representation for action recognition. Laptev [1] has introduced space-time interest points by extending the Harris detector to video. Other detection approaches are based on Gabor filters [2, 3] and on the determinant of the spatio-temporal Hessian matrix [4]. Feature descriptors range from higher order derivatives (local jets), gradient information, optical flow, and brightness information [3, 5, 6] to spatio-temporal extensions of image descriptors, such as 3D-SIFT [7], HOG3D [8], extended SURF [4], and Local Trinary Patterns [9].

However, the 2D space domain and 1D time domain in videos show different characteristics. It is, therefore, more intuitive to handle them in a different manner than to detect interest points in a joint 3D space. Tracking interest points through video sequences is a straightforward choice. Some recent methods [10, 11, 12, 13] show good results for action recognition by leveraging the motion information of trajectories. To obtain feature trajectories, either tracking techniques based on the KLT tracker [14] are used [10, 11], or SIFT descriptors between consecutive frames are matched [12]. Recently, Sun *et al.*[13] combined both approaches and added random trajectories in low density regions of both trackers in order to increase density.

Dense sampling has shown to improve results over sparse interest points for image classification [15, 16]. The same is observed for action recognition in a recent evaluation [17], where dense sampling at regular positions in space and time outperforms state-of-the-art spatio-temporal interest point detectors. In this work, we propose to sample feature points on a dense grid in each frame and track them using a state-of-the-art dense optical flow algorithm. This allows to improve the quality of the trajectories significantly over sparse tracking techniques, such as the KLT tracker. The resulting trajectories are more robust, in particular in the presence of fast irregular motions [18, 19, 20], see Figure 1.

Camera motion is very common in real-world video data such as Hollywood movies and Web videos.

To reduce the influence of camera motion on action recognition, we introduce a descriptor based on motion boundaries, initially developed in the context of human detection [21]. Motion boundaries are computed by a derivative operation on the optical flow field. Thus, motion due to locally translational camera movement is canceled out and relative motion is captured (see Figure 5). We show that motion boundaries provide a robust descriptor for action recognition that significantly outperforms existing state-of-the-art descriptors.

To evaluate our video description, we perform action classification with a bag-of-features representation and a SVM classifier [5]. Spatio-temporal pyramids are used to embed structure information and a multi-channel approach to combine the different features (trajectory shape, HOG, HOF, MBH). We evaluate the improvement of dense trajectories over KLT and SIFT trajectories. Furthermore, we compare different types of descriptors, investigate the impact of various parameters, and study the computational complexity. Experimental results on nine action datasets show a significant improvement over the state of the art.

This paper is organized as follows. Section 2 reviews related work. Section 3 presents our video representation with dense trajectories and motion boundary descriptors. The experimental setup and the datasets are described in sections 4 and 5. Experimental results are given in section 6. Section 7 concludes the paper.

The code for computing dense trajectories and descriptors is available on-line[1]. A preliminary version of this article has appeared in [20].

2 Related work

There exists a large number of approaches for extracting local spatio-temporal features in videos. Laptev [1] have introduced spatio-temporal interest points, which are an extension of the Harris detector from image to video. Interest points are local maxima of a cornerness criterion based on the spatio-temporal second-moment matrix at each video point. Dollár *et al.*[3] have proposed a cornerness function that combines a 2D Gaussian filter in space with a 1D Gabor filter in time. Bregonzio *et al.*[2] have extended this approach with 2D Gabor filters of different orientations. The spatio-temporal Hessian detector [4] relies on the determinant of the spatio-temporal Hessian matrix. Wong and Cipolla [22] have added global information to the interest point detection by applying non-negative matrix factorization (NNMF) on the entire video sequence. The locations extracted by all these approaches are sparse and detect salient motion patterns.

To describe spatio-temporal points, Schüldt *et al.*[6] use higher order derivatives (local jets). Dollár *et al.*[3] rely on descriptors based on normalized brightness, gradient, and optical flow information. Scovanner *et al.*[7] extend the popular SIFT descriptor [23] to the spatio-temporal domain, and Kläser *et al.*[8] introduce the HOG3D descriptor. Willems *et al.*[4] generalizes the image SURF descriptor [24] to the video domain by computing weighted sums of uniformly sampled responses of spatio-temporal Haar wavelets. Yeffet and Wolf [9] propose Local Trinary Patterns for videos as extension of Local Binary Patterns (LBP) [25]. Laptev *et al.*[5] combine histograms of oriented gradients (HOG) and histograms of optical flow (HOF). Their descriptors show state-of-the-art results in a recent evaluation [17].

Spatio-temporal interest points encode video information at a given location in space and time. In contrast, trajectories track a given spatial point over time and, thus, capture motion information. Messing *et al.*[11] extract feature trajectories by tracking Harris3D interest points [1] with a KLT tracker [14]. Trajectories are represented as sequences of log-polar quantized velocities and used for action classification. Matikainen *et al.*[10] extract trajectories using a standard KLT tracker, cluster the trajectories, and compute an affine transformation matrix for each cluster center. The elements of the matrix are then used to represent the trajectories. Sun *et al.*[12] compute trajectories by matching SIFT descriptors between

[1] `http://lear.inrialpes.fr/software`

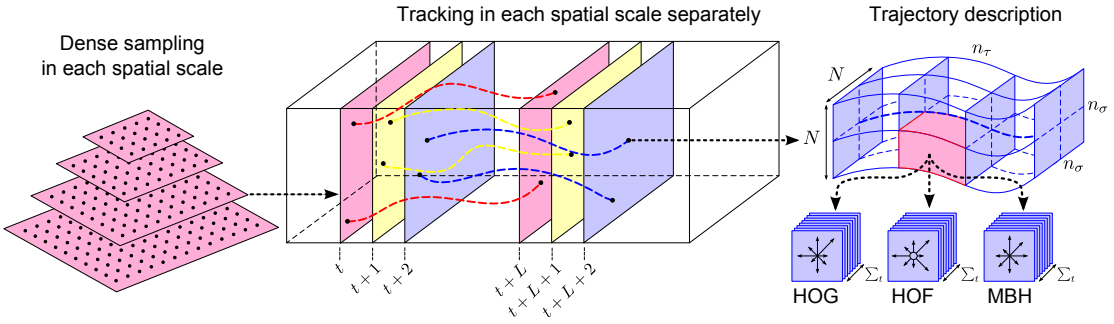


Figure 2: Illustration of our approach to extract and characterize dense trajectories. Left: Feature points are densely sampled on a grid for each spatial scale. Middle: Tracking is carried out in the corresponding spatial scale for L frames by median filtering in a dense optical flow field. Right: The trajectory shape is represented by relative point coordinates, and the descriptors (HOG, HOF, MBH) are computed along the trajectory in a $N \times N$ pixels neighborhood, which is divided into $n_\sigma \times n_\sigma \times n_\tau$ cells.

two consecutive frames. They impose a unique-match constraint among the descriptors and discarded matches that are too far apart. Actions are described with intra- and inter-trajectory statistics. Sun *et al.*[13] combine both KLT tracker and SIFT descriptor matching to extract long-duration trajectories. To assure a dense coverage with trajectories, random points are sampled for tracking within the region of existing trajectories. Spatio-temporal statistics of the trajectories are then used to discriminate different actions. Raptis and Soatto [26] track feature points in regions of interest. They compute tracklet descriptors as concatenation of HOG or HOF descriptors along the trajectories. The final descriptor is employed for action modeling and video analysis. In the experimental section, we compare to KLT and SIFT trajectories as well as to the results of [13] and [11].

Our trajectories differ from previous methods as points are sampled densely and tracked using a dense optical flow field. Dense sampling ensures a good coverage of the video with features, and optical flow improves the quality of trajectories. Dense trajectories have not been employed previously for action recognition. Somewhat related is the approach of [19], where long term trajectories are extracted using dense optical flow. In order to segment moving objects, trajectories are clustered using a pair-wise distance measure. A similar approach is proposed in [27]. The authors use dense optical flow trajectories to extract objects from video. Sand and Teller [18] investigate long range motion estimation. Videos are represented as a set of particles whose trajectories are computed from variational optical flow.

Previous works also use trajectories at the object level, e.g., for humans or vehicles. Johnson and Hogg [28] propose to track humans and model the distribution of trajectories in order to identify atypical events. Mean shift is applied to cluster object trajectories based on multiple features in [29], and clusters with less trajectories are considered as rare events. Similarly, Jung *et al.*[30] design a framework for event detection in video surveillance based on trajectory clustering of objects and 4-D histograms. Hervieu *et al.*[31] use Hidden Markov Models to capture the temporal causality of object trajectories for unexpected event detection. Wang *et al.*[32] propose a nonparametric Bayesian model for trajectory analysis and semantic region modeling in surveillance.

To take into account camera motion, Piriou *et al.*[33] define global probabilistic motion models for both the dominant image motion (assumed to be due to camera motion) and the residual image motion (related to scene motion) to recognize dynamic video content. In the context of action recognition based on local features, only a few approaches account for camera motion. Uemura *et al.*[34] segment feature tracks to separate motion characterizing actions from the dominant camera motion. Ikizler-Cinbis and Sclaroff [35] apply video stabilization by motion compensation to remove camera motion. Recently, Wu

Figure 3: Visualization of densely sampled feature points after removing points in homogeneous areas using the criterion in Eq. (1). We only show feature points in the first spatial scale. The sampling step size W is 5 pixels, which is the default setting in our experiments.

et al.[36] decompose Lagrangian particle trajectories into camera-induced and object-induced components for videos acquired by a moving camera. In contrast to these approaches, our descriptor is based on motion boundary histograms which remove constant motion and therefore reduce the influence of camera motion.

3 Dense trajectories

In this section, we present how to extract dense trajectories and compute trajectory-aligned descriptors. An overview of our approach is shown in Figure 2.

3.1 Dense sampling

We first densely sample feature points on a grid spaced by W pixels. Sampling is carried out on each spatial scale separately, see Figure 2(left). This guarantees that feature points equally cover all spatial positions and scales. Experimental results showed that a sampling step size of $W = 5$ pixels is dense enough to give good results over all datasets. There are at most 8 spatial scales in total, depending on the resolution of the video. The spatial scale increases by a factor of $1/\sqrt{2}$.

Our goal is to track all these sampled points through the video. However, in homogeneous image areas without any structure, it is impossible to track any point. We remove points in these areas. Here, we use the criterion of [37], that is points on the grid are removed, if the eigenvalues of the auto-correlation matrix are very small. We set a threshold T on the eigenvalues for each frame I as

$$T = 0.001 \times \max_{i \in I} \min(\lambda_i^1, \lambda_i^2), \tag{1}$$

where $(\lambda_i^1, \lambda_i^2)$ are the eigenvalues of point i in the image I. Experimental results showed that a value of 0.001 represents a good compromise between saliency and density of the sampled points. Sampled points of an example frame are illustrated in Figure 3. It can be seen that most points in homogeneous areas have been removed.

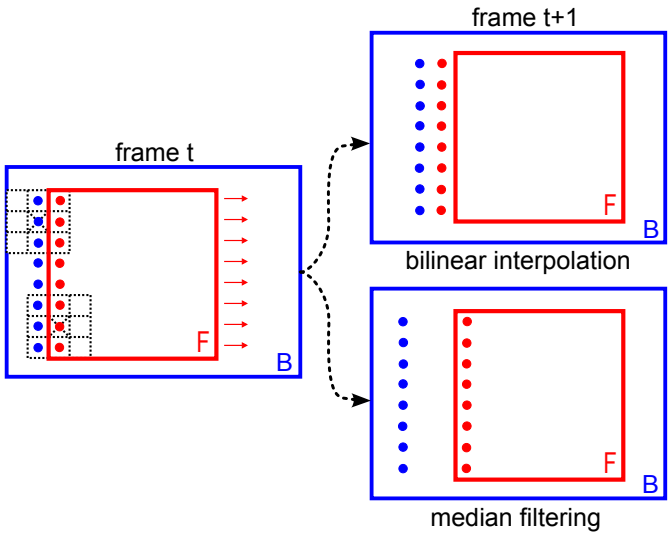


Figure 4: Comparison of bilinear interpolation and median filtering. The object is moving to the right. Pixels along the motion boundary are indicated by the blue and red dots. The blue dots belonging to the background should stay in place, whereas the red dots should follow the moving foreground object. Using bilinear interpolation, motion boundaries are blurred, and thus foreground and background motion information are confused. Median filtering allows to maintain a sharp motion boundary.

3.2 Trajectories

Feature points are tracked on each spatial scale separately. For each frame I_t, its dense optical flow field $\omega_t = (u_t, v_t)$ is computed w.r.t. the next frame I_{t+1}, where u_t and v_t are the horizontal and vertical components of the optical flow. Given a point $P_t = (x_t, y_t)$ in frame I_t, its tracked position in frame I_{t+1} is smoothed by applying a median filter on ω_t:

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M * \omega_t)|_{(x_t, y_t)}, \tag{2}$$

where M is the median filtering kernel. The size of the median filter kernel M is 3×3 pixels. As the median filter is more robust to outliers than bilinear interpolation (as used by [38]), it improves trajectories for points at motion boundaries that would otherwise be smoothed out (c.f., Figure 4).

Once the dense optical flow field is computed, points can be tracked very densely without additional cost. Another advantage of the dense optical flow is the smoothness constraints which allow relatively robust tracking of fast and irregular motion patterns, see Figure 1. To extract dense optical flow fields, we use the algorithm by [39] which embeds a translation motion model between neighborhoods of two consecutive frames. Polynomial expansion is employed to approximate pixel intensities in the neighborhood. We use the implementation from the *OpenCV* library[2].

Points of subsequent frames are concatenated to form trajectories: $(P_t, P_{t+1}, P_{t+2}, \ldots)$. As trajectories tend to drift from their initial locations during the tracking process, we limit their length to L frames in order to overcome this problem (c.f., Figure 2 (middle)). For each frame, if no tracked point is found in a $W \times W$ neighborhood, a new point is sampled and added to the tracking process so that a dense coverage of trajectories is ensured. Empirically, we set the trajectory length to $L = 15$ frames (c.f., section 6.4).

[2] `http://opencv.willowgarage.com/wiki/`

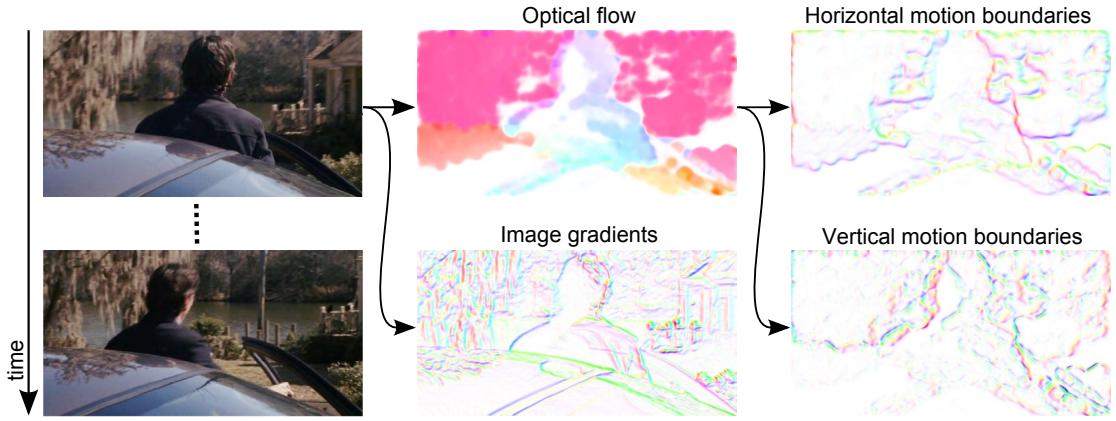


Figure 5: Illustration of the information captured by HOG, HOF, and MBH descriptors. The camera is moving from right to left, and the person is walking away from the camera. Gradient/flow orientation is indicated by color (hue) and magnitude by saturation. The optical flow (top, middle) shows constant motion in the background, which is due to the camera movements. The motion boundaries (right) encode the relative motion between the person and the background.

As static trajectories do not contain motion information, we prune them in a post-processing stage. Trajectories with sudden large displacements, most likely to be erroneous, are also removed. Such trajectories are detected, if the displacement vector between two consecutive frames is larger than 70% of the overall displacement of the trajectory.

Figure 1 shows the resulting dense trajectories and compares them to KLT and SIFT trajectories. We can observe that dense trajectories are more coherent.

3.2.1 Trajectory shape descriptor

The shape of a trajectory encodes local motion patterns. Given a trajectory of length L, we describe its shape by a sequence $(\Delta P_t, \ldots, \Delta P_{t+L-1})$ of displacement vectors $\Delta P_t = (P_{t+1} - P_t) = (x_{t+1} - x_t, y_{t+1} - y_t)$. The resulting vector is normalized by the sum of displacement vector magnitudes:

$$T = \frac{(\Delta P_t, \ldots, \Delta P_{t+L-1})}{\sum_{j=t}^{t+L-1} ||\Delta P_j||}. \tag{3}$$

In the following, we refer to this vector as *trajectory*. As we use trajectories with a fixed length of $L = 15$ frames, we obtain a 30 dimensional descriptor.

3.3 Motion and structure descriptors

Besides the trajectory shape information, we also design descriptors to embed appearance and motion information. Previous local descriptors [3, 8, 5, 7, 4] are usually computed in a 3D video volume around interest points. These representations ignore the intrinsic dynamic structures in the video. We compute descriptors within a space-time volume aligned with a trajectory to encode the motion information, Figure 2 (right). The size of the volume is $N \times N$ pixels and L frames long. To embed structure information, the volume is subdivided into a spatio-temporal grid of size $n_\sigma \times n_\sigma \times n_\tau$. We compute a descriptor (e.g., HOG, HOF or MBH) in each cell of the spatio-temporal grid, and the final descriptor is a concatenation of these descriptors. The default parameters for our experiments are $N = 32, n_\sigma = 2, n_\tau = 3$, which

showed to give best performance when cross validating on the training set of Hollywood2. We evaluate the performance of different values for this parameter in section 6.4.

3.3.1 Gradient and optical flow histograms

As shown in [17], HOG (histograms of oriented gradients) and HOF (histograms of optical flow) descriptors [5] yield excellent results on a variety of datasets in comparison with other state-of-the-art descriptors for action recognition. HOG [40] focuses on static appearance information, whereas HOF captures the local motion information.

We compute the HOG and HOF descriptors along the dense trajectories. For both HOG and HOF, orientations are quantized into 8 bins with full orientation and magnitudes are used for weighting. An additional zero bin is added for HOF (i.e., in total 9 bins) [5]. It accounts for pixels whose optical flow magnitudes are lower than a threshold. Both descriptors are normalized with their L_2 norm. The final descriptor size is 96 for HOG (i.e., $2 \times 2 \times 3 \times 8$) and 108 for HOF (i.e., $2 \times 2 \times 3 \times 9$). Figure 5 (middle) visualizes the gradient and optical flow information for HOG and HOF.

3.3.2 Motion boundary histograms

Optical flow represents the absolute motion between two frames, which contains motion from many sources, i.e., foreground object motion and background camera motion. If camera motion is considered as action motion, it may corrupt the action classification. Various types of camera motion can be observed in realistic videos, e.g., zooming, tilting, rotation, etc. In many cases, camera motion is locally translational and varies smoothly across the image plane.

Dalal *et al.*[21] proposed the motion boundary histograms (MBH) descriptor for human detection by computing derivatives separately for the horizontal and vertical components of the optical flow. The descriptor encodes the relative motion between pixels, as shown in Figure 5 (right). Since MBH represents the gradient of the optical flow, locally constant camera motion is removed and information about changes in the flow field (i.e., motion boundaries) is kept. MBH is more robust to camera motion than optical flow, and thus more discriminative for action recognition.

In this work, we employ MBH as motion descriptor for trajectories. The MBH descriptor separates optical flow $\omega = (u, v)$ into its horizontal and vertical components. Spatial derivatives are computed for each of them and orientation information is quantized into histograms. The magnitude is used for weighting. We obtain a 8-bin histogram for each component (i.e., MBHx and MBHy). Both histogram vectors are normalized separately with their L_2 norm. The dimension is 96 (i.e., $2 \times 2 \times 3 \times 8$) for both MBHx and MBHy.

Compared to video stabilization [35] and motion compensation [34], this is a simpler way to discount for camera motion. The MBH descriptor is shown to outperform significantly the HOF descriptor in our experiments, see section 6.

For both HOF and MBH descriptor computation, we reuse the dense optical flow that is already computed to extract dense trajectories. This makes our feature computation process more efficient. A detailed analysis of the complexity is given in section 6.5.

4 Experimental setup

In this section, we first introduce our baseline methods for trajectory extraction. We then detail the bag-of-features representation as used in our experiments and finally present spatio-temporal pyramids.

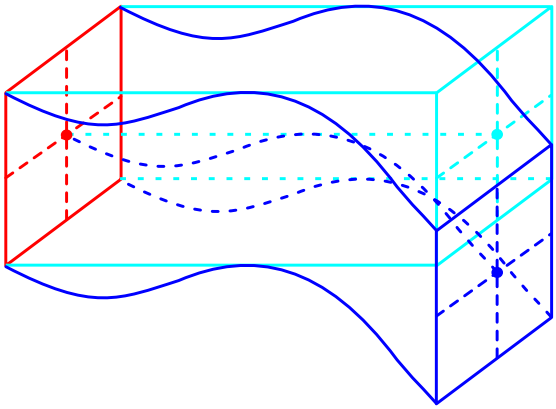


Figure 6: Illustration of the difference between a dense cuboid and a dense trajectory. Both approaches start at the same position (the red dot), but then continue differently.

4.1 Baseline trajectories

To quantify the improvement obtained with our dense trajectories, we compare to three baseline trajectories in our experimental results: KLT trajectories [10, 11, 13], SIFT trajectories [12, 13], as well as dense cuboids. Details are presented in the following.

4.1.1 KLT trajectories

A standard KLT tracker [14] is employed to construct KLT trajectories. More specifically, about 100 interest points are detected in each frame and are added to the tracker – this is somewhat denser than space-time interest points [17]. The points are tracked through the video for $L = 15$ frames. Then they are removed and replaced by new interest points. This is identical to the procedure used for dense trajectories. We use the same descriptors for KLT trajectories, i.e., the trajectory shape is represented by normalized displacement vectors and HOG, HOF, as well as MBH descriptors are extracted along the trajectories. We use the *OpenCV* implementation of the KLT tracker. We also examine the impact of an increasing number of KLT trajectories obtained by decreasing the threshold to extract interest points.

4.1.2 SIFT trajectories

To extract SIFT trajectories, we first extract SIFT interest points. The best match in the next frame is the point with the smallest Euclidean distance between the SIFT descriptors[3] within a neighborhood. We set the contrast threshold for the SIFT detector to 0.004, which is one order lower than the default setting, and makes sure that there are enough SIFT interest points for matching. We further decrease the threshold to extract more SIFT points to obtain additional trajectories and examine the impact on the recognition performance (c.f., Figure 9).

A visualization of SIFT trajectories is represented in Figure 1. Unlike the KLT trajectories, SIFT trajectories are very robust to shot boundaries, i.e., most trajectories crossing the shot boundaries are removed. This is due to the descriptive power of SIFT. However, we observed that SIFT trajectories are not able to model subtle motion changes in videos. This is presumably due to the fact that the descriptor is based on spatial histograms, which are not well localized [41]. Moreover, SIFT interest points are only detected on a set of discrete spatial positions, whereas dense optical flow can provide subpixel accuracy.

[3]The code of SIFT detector and descriptor is from `http://blogs.oregonstate.edu/hess/code/sift/`

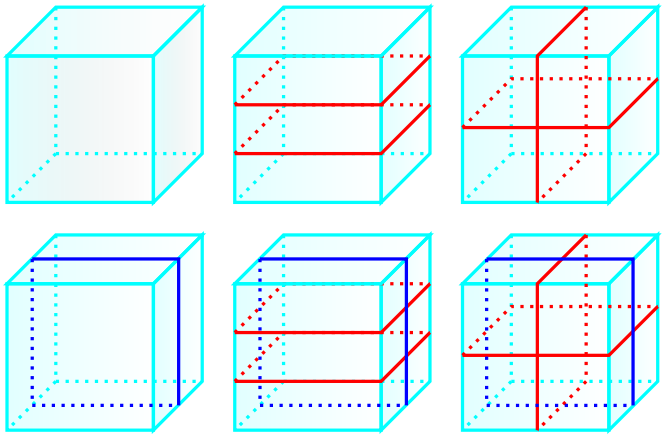


Figure 7: The spatio-temporal grids used in our experiments: $h1 \times v1 \times t1$, $h3 \times v1 \times t1$, $h2 \times v2 \times t1$, $h1 \times v1 \times t2$, $h3 \times v1 \times t2$, $h2 \times v2 \times t2$.

4.1.3 Dense cuboids

We compare to dense cuboids in order to demonstrate the benefit of descriptors aligned to trajectories. To this end, feature points are sampled in the same way as for dense trajectories. This guarantees that both dense cuboids and dense trajectories have the same number of features at the same positions. The only difference is that feature points are not tracked for dense cuboids. As shown in Figure 6, descriptors are extracted within a straight 3D block. HOG, HOF and MBH descriptors are computed in this block using the same parameters as for dense trajectories.

4.2 Bag of features

We apply the standard bag-of-features approach to evaluate our dense trajectory features as well as the three baseline trajectories. We first construct a codebook for each descriptor (trajectory, HOG, HOF, MBHx, MBHy)[4] and trajectory type separately. We fix the number of visual words per descriptor to 4000 which has shown to empirically give good results for a wide range of datasets. To limit the complexity, we cluster a subset of 100,000 randomly selected training features using k-means. To increase precision, we initialize k-means 8 times and keep the result with the lowest error. Descriptors are assigned to their closest vocabulary word using Euclidean distance. The resulting histograms of visual word occurrences are used as video representations.

For classification we use a non-linear SVM with an RBF-χ^2 kernel [5]. Different descriptors are combined in a multi-channel approach [42]:

$$K(x_i, x_j) = exp(-\sum_c \frac{1}{A^c} D(x_i^c, x_j^c)), \tag{4}$$

where $D(x_i^c, x_j^c)$ is the χ^2 distance between video x_i and x_j with respect to the c-th channel. A^c is the mean value of the χ^2 distances between the training samples for the c-th channel. In the case of multi-class classification, we use a *one-against-rest* approach and select the class with the highest score.

[4] Note that splitting MBH into MBHx and MBHy results in a slightly better performance.

4.3 Spatio-temporal pyramids

We add structure information to the bag of features using spatio-temporal pyramids [5, 43], an extension of spatial pyramids for images [44]. We use in our experiments six different spatio-temporal grids. For the spatial domain we use the entire spatial block $h1 \times v1$, a subdivision into three horizontal stripes $h3 \times v1$ and a 2×2 spatial grid $h2 \times v2$. For the temporal domain we use the entire duration $t1$ as well as a subdivision into 2 temporal blocks $t2$. Figure 7 illustrates our six grids, $h1 \times v1 \times t1$, $h3 \times v1 \times t1$, $h2 \times v2 \times t1$, $h1 \times v1 \times t2$, $h3 \times v1 \times t2$, $h2 \times v2 \times t2$. For each cell of the grid, a separate bag-of-features histogram is computed. The video is, then, represented as concatenation of the cell histograms. We use each grid structure as a separate channel and combined them using Eq. (4). In total, we have 30 different channels (i.e., 6 grid structures $\times 5$ descriptor types) to represent a video.

5 Datasets

This section describes the datasets we use and the experimental protocols for these datasets. We extensively evaluate our dense trajectory features on nine action datasets, i.e., KTH, YouTube, Hollywood2, UCF sports, IXMAS, UIUC, Olympic Sports, UCF50 and HMDB51, see Figure 8. These datasets are collected from various sources, e.g., controlled experimental settings, Hollywood movies, Web videos, TV sports, etc. Thus, we investigate the performance of our approach on diverse datasets with different resolutions, viewpoints, illumination changes, occlusion, background clutter, irregular motion, etc. In total, we evaluate over 20,000 video sequences and 200 action classes.

The **KTH** dataset[5] [6] consists of six human action classes: walking, jogging, running, boxing, waving and clapping. Each action is performed several times by 25 subjects. The sequences were recorded in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes and indoors. The background is homogeneous and static in most sequences. In total, the data consists of 2,391 video samples. We follow the original experimental setup of the authors, i.e., dividing the samples into test set (9 subjects: 2, 3, 5, 6, 7, 8, 9, 10, and 22) and training set (the remaining 16 subjects). As in the initial paper [6], we train and evaluate a multi-class classifier and report average accuracy over all classes.

The **YouTube** dataset[6] [45] contains 11 action categories: basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. This dataset is challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background and illumination conditions. The dataset contains a total of 1,168 sequences. We follow the original setup [45], using Leave-One-Out Cross-Validation for a pre-defined set of 25 groups. Average accuracy over all classes is reported as the performance measure[7]

The **Hollywood2** dataset[8] [46] has been collected from 69 different Hollywood movies. There are 12 action classes: answering the phone, driving car, eating, fighting, getting out of car, hand shaking, hugging, kissing, running, sitting down, sitting up, and standing up. In our experiments, we use the clean training set. In total, there are 1,707 video sequences divided into a training set (823 sequences) and a test set (884 sequences). Training and test sequences come from different movies. The performance is evaluated by computing the average precision (AP) for each action class and reporting the mean AP over all classes (mAP) as in [46].

[5] `http://www.nada.kth.se/cvap/actions/`

[6] `http://www.cs.ucf.edu/~liujg/YouTube_Action_dataset.html`

[7] Note that here we use the same dataset as [45], whereas in [20] we used a different version. This explains the difference in performance on the YouTube dataset.

[8] `http://lear.inrialpes.fr/data`

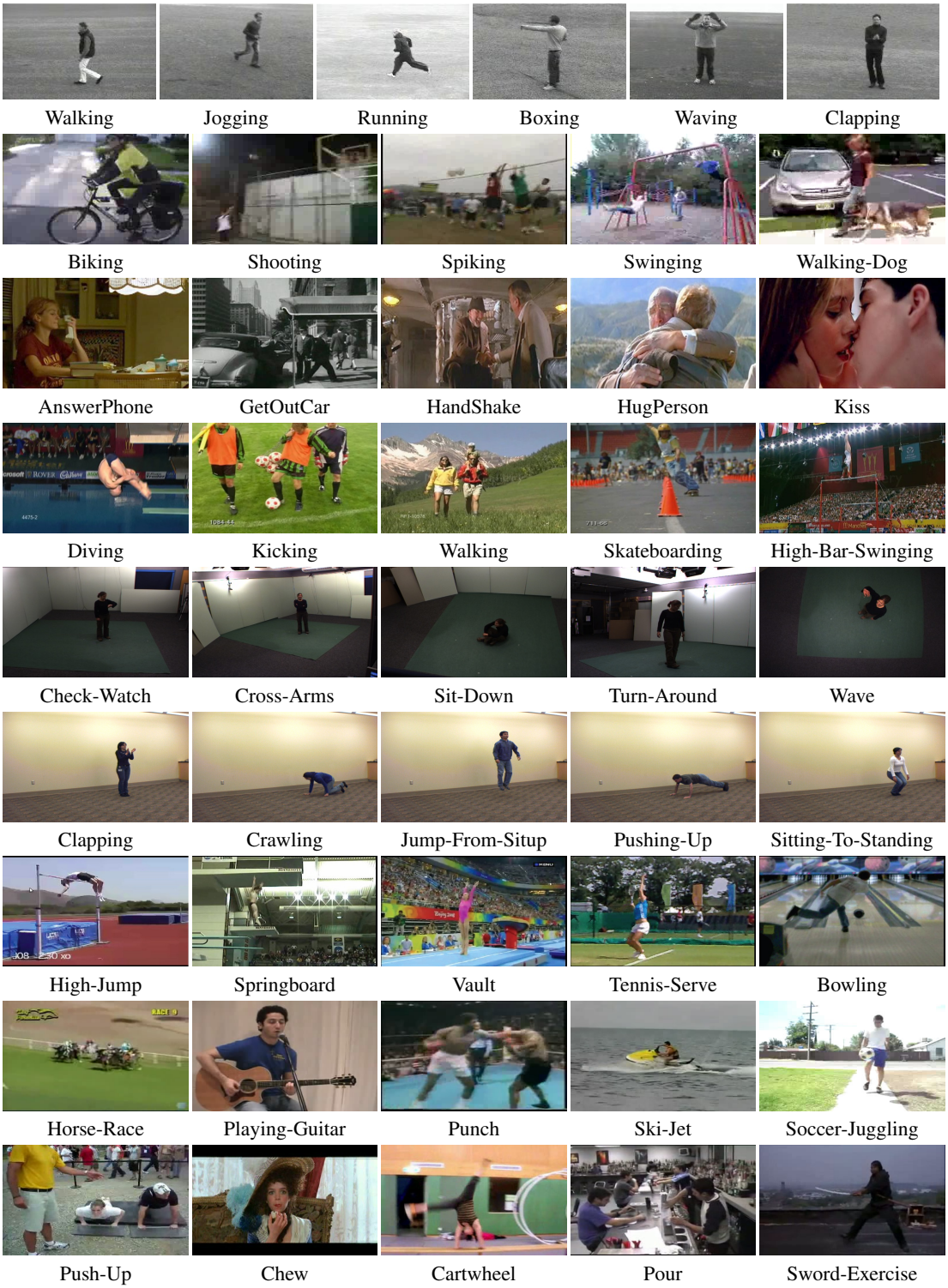


Figure 8: Sample frames from the nine action recognition datasets used in our experiments. From top to bottom: KTH, YouTube, Hollywood2, UCF sports, IXMAS, UIUC, Olympic Sports, UCF50 and HMDB51.

The **UCF sports** dataset[9] [47] contains ten human actions: swinging (on the pommel horse and on the floor), diving, kicking (a ball), weight-lifting, horse-riding, running, skateboarding, swinging (at the high bar), golf swinging and walking. The dataset consists of 150 video samples which show a large intra-class variability. To increase the amount of data samples, we extend the dataset by adding a horizontally flipped version of each sequence to the dataset. Similar to the KTH dataset, we train a multi-class classifier and report average accuracy over all classes. We use the Leave-One-Out setup, i.e., testing on each original sequence while training on all the other sequences together with their flipped versions. Note that the flipped version of the tested sequence is removed from the training set as in [17].

IXMAS[10] [48] is a dataset recorded by cameras from five different viewpoints, as shown in the fifth row of Figure 8. It has 11 action classes, i.e., check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick and pick up. All actions are repeated three times by each of the ten actors and recorded simultaneously from five views, which results in 1,650 sequences in total. We apply Leave-One-Actor-Out Cross-Validation as recommended in the original paper [48], and use samples from all five views for training and testing. Average accuracy over all classes is used as performance measure.

The **Olympic Sports** dataset[11] [49] consists of athletes practicing different sports, which are collected from YouTube and annotated using Amazon Mechanical Turk. There are 16 sports actions: high-jump, long-jump, triple-jump, pole-vault, basketball lay-up, bowling, tennis-serve, platform, discus, hammer, javelin, shot-put, springboard, snatch, clean-jerk and vault, represented by a total of 783 video sequences. We use 649 sequences for training and 134 sequences for testing as recommended by the authors. Like UCF sports, this dataset has rich scene context information, which is very helpful for recognizing sports actions. We report mean average precision over all classes (mAP) as in [49].

The **UIUC** dataset[12] [50] is recorded in a controlled experimental setting with clean background and fixed cameras. There are 14 action classes, i.e., walking, running, jumping, waving, jumping jacks, clapping, jump from situp, raise one hand, stretching out, turning, sitting to standing, crawling, pushing up and standing to sitting. In total, it contains 532 videos performed by 8 persons. We use Leave-One-Person-Out Cross-Validation following the original paper, and report average accuracy over all classes as performance measure.

The **UCF50** dataset[13] [51] has 50 action categories, consisting of real-world videos taken from the YouTube website. This dataset can be considered as an extension of the YouTube dataset. 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 same Leave-One-Group-Out Cross-Validation as for the YouTube dataset and report average accuracy over all classes.

The **HMDB51** dataset[14] [52] is collected from a variety of sources ranging from digitized movies to YouTube videos. It contains simple facial actions, general body movements and human interactions. In total, there are 51 action categories and 6,766 video sequences. We follow the original protocol using three train-test splits [52]. 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 the dataset includes both the original videos and their stabilized version. With the exception of Table 3, we report results for the original videos.

[9] `http://server.cs.ucf.edu/~vision/data.html`
[10] `http://4drepository.inrialpes.fr/public/viewgroup/6`
[11] `http://vision.stanford.edu/Datasets/OlympicSports/`
[12] `http://vision.cs.uiuc.edu/projects/activity/`
[13] `http://server.cs.ucf.edu/~vision/data/UCF50.rar`
[14] `http://serre-lab.clps.brown.edu/resources/HMDB/`

	Datasets	KTH	YouTube	Hollywood2	UCF sports	IXMAS	UIUC	Olympic Sports	UCF50	HMDB51
Dense trajectories	Trajectory	89.8%	67.5%	47.8%	75.4%	87.8%	**98.1%**	60.5%	67.2%	28.0%
	HOG	87.0%	72.6%	41.2%	**84.3%**	82.3%	97.9%	63.0%	68.0%	27.9%
	HOF	93.3%	70.0%	50.3%	76.8%	90.8%	97.7%	58.7%	68.2%	31.5%
	MBH	**95.0%**	**80.6%**	**55.1%**	84.2%	**91.8%**	97.1%	**71.6%**	**82.2%**	**43.2%**
	Combined	**94.2%**	**84.1%**	**58.2%**	**88.0%**	**93.5%**	**98.4%**	**74.1%**	**84.5%**	**46.6%**
KLT trajectories	Trajectory	88.4%	58.2%	46.2%	72.8%	75.2%	96.9%	48.1%	52.8%	22.0%
	HOG	84.0%	71.0%	41.0%	80.2%	74.4%	97.3%	50.5%	57.4%	22.2%
	HOF	92.4%	64.1%	48.4%	72.7%	84.1%	97.2%	49.2%	57.9%	23.7%
	MBH	93.4%	72.9%	48.6%	78.4%	82.5%	96.6%	59.4%	71.1%	33.7%
	Combined	93.4%	79.5%	54.6%	82.1%	91.8%	**98.1%**	65.5%	78.1%	40.8%
SIFT trajectories	Trajectory	44.6%	47.3%	39.9%	55.7%	34.7%	87.1%	47.4%	43.6%	17.6%
	HOG	59.1%	59.6%	33.3%	74.2%	45.0%	86.2%	52.8%	53.4%	21.2%
	HOF	78.8%	55.4%	38.7%	69.9%	63.0%	91.4%	45.0%	52.9%	22.4%
	MBH	79.5%	64.5%	40.6%	72.1%	58.9%	90.3%	54.5%	62.7%	26.3%
	Combined	84.9%	73.2%	45.4%	77.9%	67.9%	90.8%	61.5%	71.8%	32.5%
Dense cuboids	HOG	77.0%	70.3%	43.3%	80.2%	79.4%	96.5%	62.7%	64.4%	25.2%
	HOF	90.5%	68.3%	48.0%	77.8%	90.3%	97.1%	57.3%	65.9%	29.4%
	MBH	**93.9%**	78.4%	52.1%	83.2%	91.2%	97.6%	66.6%	78.3%	40.9%
	Combined	93.1%	**81.4%**	**55.2%**	**85.5%**	**92.8%**	97.3%	**69.3%**	**80.2%**	**43.1%**

Table 1: Comparison of different descriptors and methods for extracting trajectories on nine datasets. We report mean average precision over all classes (mAP) for Hollywood2 and Olympic Sports, average accuracy over all classes for the other seven datasets. The three best results for each dataset are in bold.

6 Experimental results

This section evaluates our dense trajectories and motion-boundary descriptors on nine datasets. We first discuss the performance of different descriptors for dense trajectories in section 6.1. In section 6.2, we compare dense trajectories with three baselines, i.e., KLT trajectories, SIFT trajectories and dense cuboids. Section 6.3 presents the results of dense trajectories using different optical flow algorithms. The parameters of our approach are evaluated in section 6.4 and the computational complexity is analyzed in section 6.5. Finally, we compare to state-of-the-art results in section 6.6.

We use the same default parameters on all datasets. Unless stated otherwise, we set $N = 32, n_\sigma = 2, n_\tau = 3$, and fix the trajectory length L to 15 frames. The sampling step size is $W = 5$ pixels.

6.1 Comparison of different descriptors

The different descriptors are compared in Table 1. We report the performance of each individual descriptor (i.e., trajectory, HOG, HOF and MBH), and the overall combination using the multi-channel approach (c.f., Eq. (4)). The results for MBH are obtained by combining MBHx and MBHy channels.

The trajectory descriptor gives surprisingly good results by itself, (89.8% on KTH, 87.8% on IXMAS, 98.1% on UIUC and 47.8% on Hollywood2). Trajectory shape information is able to outperform HOG on KTH, IXMAS, UIUC and Hollywood2. As the first three datasets all have clean background, tracking is much easier and thus carries more discriminative information. HOG does not perform well on

Datasets	KTH	YouTube	Hollywood2	UCF sports	IXMAS	UIUC	Olympic Sports	UCF50	HMDB51
MBHx	93.8%	76.1%	48.5%	83.8%	86.5%	95.6%	67.4%	75.0%	32.2%
MBHy	94.2%	76.0%	52.9%	82.9%	89.3%	95.6%	67.4%	77.9%	38.8%
MBH	95.0%	80.6%	55.1%	84.2%	91.8%	97.1%	71.6%	82.2%	43.2%

Table 2: Results of MBHx, MBHy and MBH descriptors for dense trajectories on all nine datasets. MBHx and MBHy are combined using the multi-channel approach.

Datasets	Trajectory	HOG	HOF	MBH	Combined
original	28.0%	27.9%	31.5%	43.2%	46.6%
stabilized	34.0%	30.1%	39.5%	44.9%	50.8%

Table 3: Descriptor comparison on HMDB51 for the original videos and their stabilized version provided by the authors [52].

Hollywood2 as complex cluttered background degrades its discriminative power.

One may expect the HOF descriptor to outperform HOG as motion seems intuitively more discriminative than static appearance for action recognition. However, HOG reports better results on three datasets: YouTube, UCF sports and Olympic Sports. In fact, all three datasets contain a significant amount of sports actions. UCF sports and Olympic Sports are both by definition sports datasets, whereas a large part of actions in YouTube are sports-related, e.g., basketball shooting, biking, golf swinging, etc. Spatial context is very informative for sports actions as they usually involve specific equipments and particular environments. HOG is designed to encode this static context information and, thus, is more suitable for sports actions. Additionally, the performance of HOF can be corrupted by camera motion for these datasets.

MBH consistently outperforms the other descriptors on all the datasets except UIUC. The improvement over HOF is most significant on real-world datasets. For instance, MBH is 14% better than HOF on UCF50. We can also observe an improvement of over 10% on Olympic Sports, YouTube and HMDB51. This indicates that suppressing camera motion is advantageous for videos from real-world datasets as they are often collected by hand-held cameras, e.g., YouTube and UCF50. For datasets with fixed cameras, e.g., IXMAS and UIUC, the results for MBH and HOF are very similar.

We show the results for MBHx and MBHy separately in Table 2. Generally, MBHx and MBHy show similar performance. Interestingly, MBHy outperforms MBHx by 4.5% on Hollywood2. A possible explanation is that this dataset contains several actions, such as sit up, sit down and stand up, which are dominant in the vertical direction. Similar observations hold for IXMAS, UCF50 and HMDB51.

In general, the combination of all the descriptors improves the final performance significantly, see Table 1. Results are 3% better than the single best descriptor on YouTube, Hollywood2, UCF sports and HMDB51. For datasets from controlled settings, the improvement is less significant. On KTH, MBH alone is even slightly better (i.e., 95.0%) than combining all the descriptors together.

We investigate the impact of video stabilization on the video description for HMDB51, see Table 3. We can observe that the performance of the trajectory descriptor improves by 6% as irrelevant trajectories due to camera motion are largely removed in stabilized videos. The improvement for HOF is even more significant, i.e., 8%. This can be explained by its lack of robustness to camera motion. The performance of MBH is similar in both cases, which demonstrates its robustness to camera motion. The HOG descriptor improves only marginally, as it mainly represents static appearance.

We also compare descriptors computed at space-time interest points extracted with the Harris3D de-

Datasets	Descriptors of [5]			Our descriptors			
	HOG	HOF	Combined	HOG	HOF	MBH	Combined
YouTube	61.7%	61.3%	69.2%	62.5%	62.1%	70.9%	76.2%
Hollywood2	41.2%	43.6%	47.7%	40.4%	44.9%	47.2%	51.9%

Table 4: Descriptor comparison on YouTube and Hollywood2. Descriptors are computed at space-time interest point locations [5].

Dataset	Trajectory	Sun *et al.*[13]	Messing *et al.*[11]
KTH	89.8%	86.8%	74.0%

Table 5: Comparing the performance of trajectory descriptor obtained by different methods on the KTH dataset, as reported in the respective papers. Trajectory refers to our trajectory descriptor obtained for dense trajectories.

tector [1]. Table 4 compares the HOG and HOF descriptors of [5] to our implementation of HOG and HOF as well as MBH. For our descriptors, we take the positions of Harris3D interest points, and then compute descriptors in the 3D patches around these positions with the same parameters as for the dense trajectories, i.e., the size of the 3D patch is 32×32 pixels and 15 frames. The performance of HOG and HOF descriptors for both methods are comparable as to be expected. MBH outperforms all other descriptors by a large margin and improves significantly over HOF.

6.2 Comparison to baseline trajectories

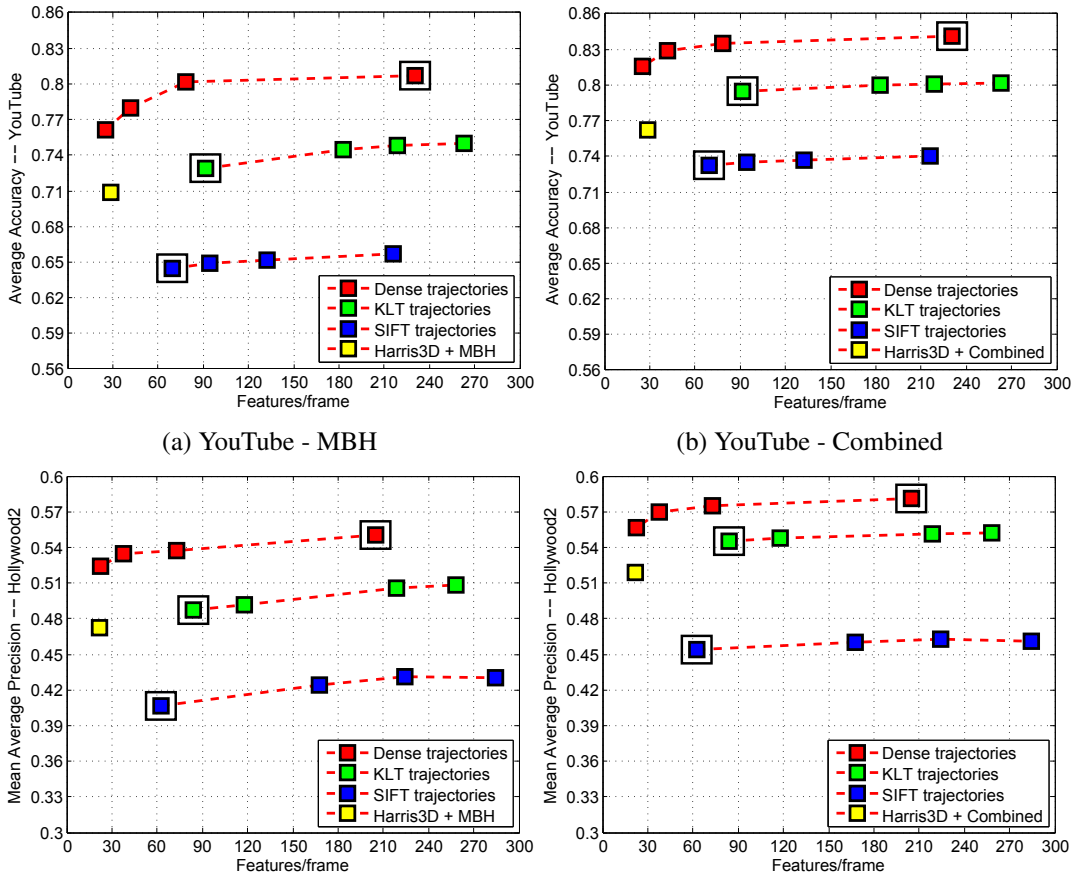
We use the default parameters for characterizing the baselines trajectories, e.g., $N = 32, n_\sigma = 2, n_\tau = 3$, and a trajectory length L of 15 frames for KLT and SIFT trajectories. The temporal length of dense cuboids is also 15 frames, with the same initial points as for the dense trajectories. Results are given in Table 1.

We first compare the performance of the trajectory descriptor for KLT, SIFT and dense trajectories. On all nine datasets, the trajectory descriptor obtained with dense trajectories report the best results. The largest improvement over KLT trajectories is 14.4% on UCF50. On YouTube, IXMAS and Olympic Sports, we can observe an improvement of over 9%. This indicates that our dense trajectories capture better the dynamics in videos. SIFT trajectories perform the worst among the three trajectories, especially on datasets with clean background, e.g., KTH and IXMAS. This can be explained by the fact that SIFT interest points are rather sparse on these datasets and result in non-smooth trajectories.

Table 5 compares the performance of our trajectory shape descriptor with [13] and [11] on the KTH dataset[15]. We can observe that our trajectory descriptor (i.e., normalized point coordinates) improves over these two trajectory based methods. This may be due to the quality of our trajectories, i.e., tracking with dense optical flow significantly improves the quality of trajectories. Furthermore, our dense sampling strategy guarantees that features points are distributed equally over all spatial scales and positions, which could explain the significant difference with [11].

The improvement of other descriptors is also substantial as shown in Table 1. The HOF descriptor for dense trajectories is 9.5% better than HOF computed on KLT trajectories on Olympic Sports, whereas MBH for dense trajectories outperforms KLT trajectories by 11.1% on UCF50. In comparison with SIFT

[15]Note that we only consider the performance of the trajectory itself. Other information, such as gradient or optical flow, is not included.



(a) YouTube - MBH (b) YouTube - Combined

(c) Hollywood2 - MBH (d) Hollywood2 - Combined

Figure 9: Performance of dense, KLT and SIFT trajectories for a varying number of features per frame. On the left the results for the MBH descriptor and on the right for a combination of trajectory, HOG, HOF and MBH.

trajectories, the performance gain using dense trajectories is even larger. Overall this suggests that the quality of trajectories has a strong influence on the performance of trajectory-aligned descriptors.

We evaluate the difference of descriptors aligned to trajectories in comparison to dense spatio-temporal cuboids. Typically, the improvement of dense trajectories is around 3% for a single descriptor. The largest improvement is 10% for the HOG descriptor on KTH. In a few cases, the descriptors using dense cuboids show slightly better results than those for dense trajectories, e.g., HOF on UCF Sports (1%). In total, descriptors aligned with trajectories are superior to those computed for straight 3D blocks.

The results for combined descriptors on all trajectory types and dense cuboids show that dense trajectories yield overall the best performance on all nine datasets. The improvement is over 3% on Hollywood2, Olympic Sports, UCF50 and HMDB51. Interestingly, dense representations (i.e., dense trajectories and dense cuboids) consistently outperform sparse representations (i.e., KLT trajectories and SIFT trajectories) on YouTube, UCF sports, Olympic Sports, UCF50 and HMDB51. This shows that a high number of features improve the results, especially for sports-related datasets where context information can play an important role. For example, dense trajectories improve by 8.6% on Olympic Sports over KLT trajectories. However, for datasets collected in controlled experimental settings, e.g., KTH, IXMAS

Overlaid image | Farnebäck [39] | Brox and Malik [41]

Figure 10: Examples from Hollywood2 (top) and YouTube (bottom). The left column shows the overlaid image of two consecutive frames. The middle and right columns are the visualization of two optical flow methods [39] and [41].

and UIUC, the performance gain is less significant.

We also evaluate the impact of the feature density on the performance of KLT, SIFT and dense trajectories, see Figure 9. The results obtained with the standard parameters and the corresponding average number of features per frame are indicated by a black rectangle. For KLT and SIFT trajectories we gradually decrease the thresholds for extracting features to increase their numbers per frame. For dense trajectories we vary the sampling stride W (i.e., 5, 8, 11, 14 pixels) to obtain fewer feature points. We also include the results for Harris3D and our descriptors from Table 4.

We can observe that increasing the number of features for KLT and SIFT increases the performance slightly and then saturates. The performance gain between the default parameters and the parameters for which the feature number is comparable to the standard dense trajectories is around 1% for both KLT and SIFT trajectories. For a comparable number of features, dense trajectories outperform KLT trajectories by around 5% (3%) for MBH (combined) results. The improvement over SIFT trajectories is around 12% (10%).

6.3 Comparison of different optical flow algorithms

Our implementation of dense trajectories uses the optical flow algorithm of [39], which represents a good compromise between speed and accuracy. In this section, we compare to a state-of-the-art optical flow algorithm, the large displacement optical flow (LDOF) from [41]. We replace Farnebäck's optical flow with LDOF, and keep everything else identical. To extract LDOF we use the binary code from the author's website [16] with the default parameters.

The results are given in Table 6. Surprisingly, the overall performance of the two optical flow algorithms is similar. The result of the MBH descriptor from LDOF is somewhat better for the YouTube dataset and slightly worse for Hollywood2. One possible explanation is that we use the default parameters for LDOF, which may not be optimal for realistic Hollywood movies. Another possible explanation is that [39] estimates optical flow locally and captures fine details in textured areas, whereas LDOF includes global smoothness constraints and preserves motion boundaries, see figure 10. Please note that LDOF is computationally expensive; its run time is significantly higher than that of [39].

[16] `http://lmb.informatik.uni-freiburg.de/resources/binaries/pami2010Linux64.zip`

Methods	YouTube					Hollywood2				
	Trajectory	HOG	HOF	MBH	Combined	Trajectory	HOG	HOF	MBH	Combined
Farnebäck [39]	67.5%	72.6%	70.0%	80.6%	84.1%	47.8%	41.2%	50.3%	55.1%	58.2%
Brox and Malik [41]	64.5%	71.9%	65.7%	83.6%	84.9%	43.7%	40.8%	46.1%	54.3%	57.2%

Table 6: Comparing the optical flow algorithms of [39] and [41] for extracting our dense trajectories. Results are reported on the YouTube and Hollywood2 datasets.

6.4 Evaluation of trajectory parameters

In this section, we evaluate the impact of the parameters on dense trajectories. We report results for Hollywood2 and YouTube. We study the impact of trajectory length, sampling step size, neighborhood size, cell grid structure, spatial scale number and refresh rate (the frame rate for which we sample feature points in time). We carry out the evaluation for one parameter at a time, and fix the other parameters to the default values, i.e., trajectory length $L = 15$, sampling step size $W = 5$, neighborhood size $N = 32$, cell grid structure $n_\sigma = 2, n_\tau = 3$, the number of spatial scale $S = 8$ and refresh rate $R = 1$.

Results for various trajectory lengths are shown in Figure 11(a). For both datasets, increasing the length L improves the performance up to $L = 20$ frames. Trajectories need to have a minimum length in order to encode enough motion information. However, trajectories longer than 20 frames decrease the results, as they have a higher chance to drift from the initial position during the tracking process or to cross shot boundaries. We observe best results with a trajectory length of 15 or 20 frames.

With respect to the sampling step size W, Figure 11(b) presents the results for $W = 2$ pixels to $W = 20$ pixels. The performance increases with a higher sampling density. This is also consistent with dense sampling at regular positions where more features in general improve the results up to a point [17]. For a step size of 2 (5) pixels, we report 58.9% (58.2%) on Hollywood2 and 84.4% (84.1%) on YouTube. A sampling stride of 2 pixels samples every other pixel which significantly increases the computational complexity, see section 6.5. $W = 5$ pixels offers a good trade-off between speed and accuracy.

Results are relatively stable with regard to the neighborhood size N, see Figure 11(c). On Hollywood2, the performance is comparable for values between $N = 24$ and $N = 48$ pixels. This is probably due to the descriptors which are strongly overlapping and thus form an over complete representation of the video. The best result on YouTube is 84.7% with a neighborhood size of 40 pixels.

Divisions of trajectory volumes into cells improve the results on both Hollywood2 and YouTube. In particular, the performance increases significantly from a spatial cell grid $n_\sigma = 1$ to $n_\sigma = 2$, see Figure 11(d). Further increasing in the number of cells, i.e., beyond $n_\sigma = 2, n_\tau = 3$, does not yield better results. As best results, we report 58.5% on Hollywood2 with a $2 \times 2 \times 2$ cell grid and 84.1% on YouTube with a $2 \times 2 \times 3$ cell grid.

Multiple spatial scales have been reported to improve the performance of local image descriptors. The same finding applies to our descriptors, see Figure 11(e). $S = 1$ refers to only the original image scale. As S increases, further scales are added. The improvement on Hollywood2 is over 2% from 56.1% to 58.2%, whereas on YouTube it is 1.5% from 82.6% to 84.1%. For both datasets the performance saturates at $S = 5$ or $S = 6$ spatial scales.

The refresh rate R controls the frame rate, at which feature points are sampled in time. With the default setting, we sample new feature points in each frame. This may not be necessary for videos with high frame rate. We show results for different refresh rates R in Figure 11(f). This parameter is more sensitive for Hollywood2 than for YouTube. Movies often contain fast motion patterns, especially in action movies. It is then advantageous to sample feature points in each frame in order to capture more information. On the other hand, YouTube is very robust to the refresh rate R, the performance hardly changes up to a rate of $R = 7$. This may be due to the temporal redundancy in YouTube videos and the

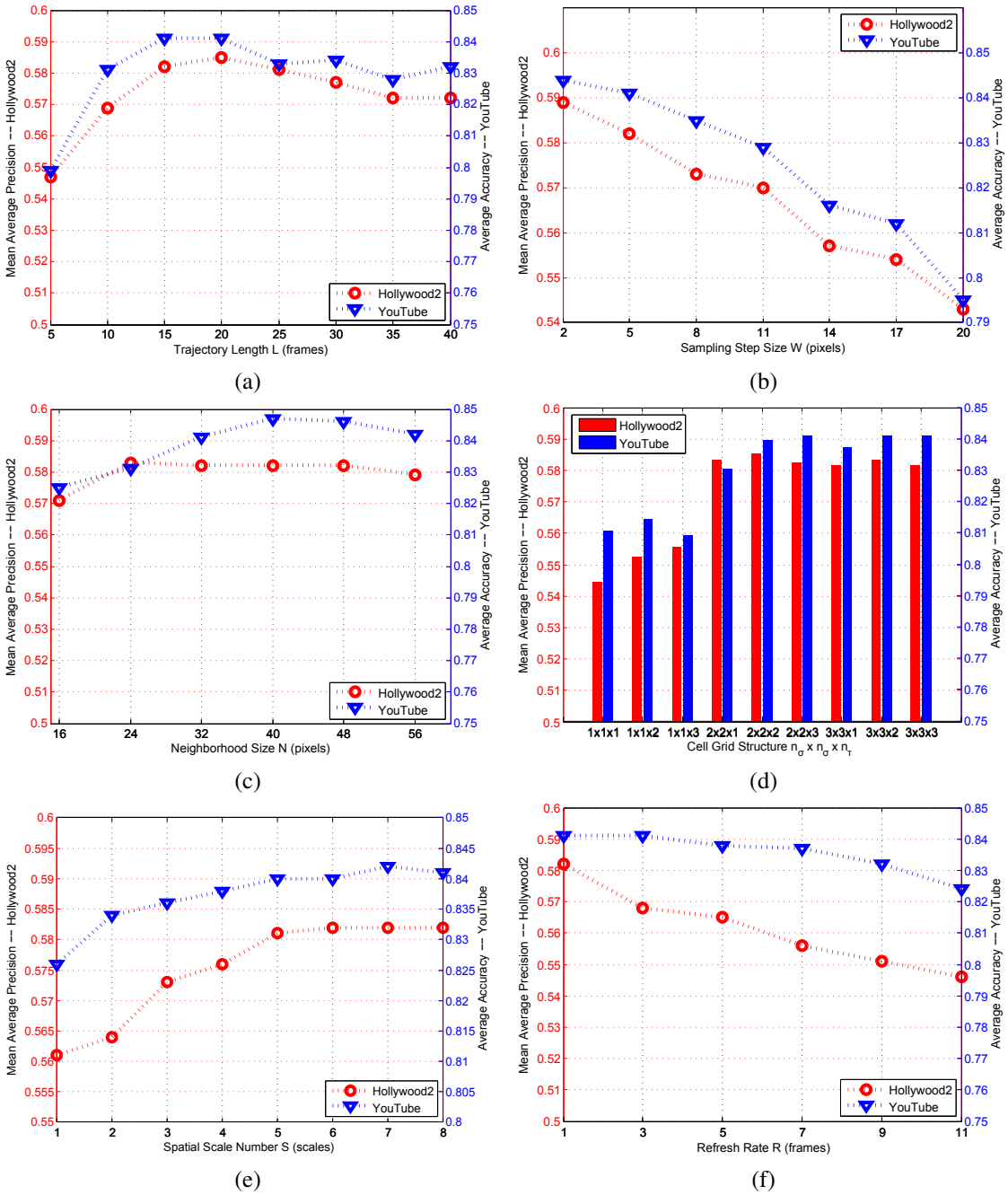


Figure 11: Evaluation of the parameters on the Hollywood2 and YouTube datasets: (a) trajectory length, (b) sampling step size, (c) neighborhood size, (d) cell grid structure, (e) spatial scale number and (f) refresh rate.

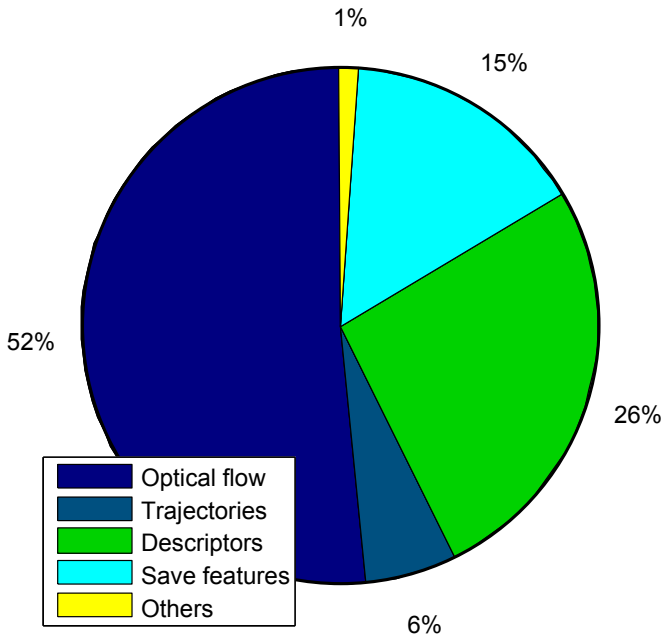


Figure 12: The percentages of the time spent on the major steps of computing dense trajectories using the default parameter setting.

missing shot boundaries.

6.5 Computational complexity

To analyze the computational complexity of the feature extraction, we compute dense trajectories for the 43 training video clips from the movie “American beauty" of the Hollywood2 dataset. The resolution is 528×224 pixels, and the 43 video clips correspond to a total of approximately $15,000$ frames. We report run time (frames per second) and the number of features per frame in Figure 13. As a comparison, we also compute STIP (Harris3D+HOGHOF) features with the widely used STIP toolbox [17] on the same videos. The run-time is obtained on a Dell server with a 2.6 GHz quad-core Opteron CPU and 8GB RAM. We do not parallelize our code and only use a single core of the CPU.

Figure 12 analyzes the percentage of time spent on each step of computing dense trajectories for the default parameters. The computation of dense optical flow consumes most of the time with 52%. Once the flow field is computed, tracking feature points is relatively inexpensive, it only takes 6% of the total computation time. Descriptor computation is the second most time-consuming step (26%) with all four descriptors (Trajectory, HOG, HOF and MBH) being extracted. If only MBH descriptors are extracted, the descriptor computation time is reduce by 46%. Note that the descriptor computation reuses the optical flow field which is only computed once. Finally, storing features to disk also takes time since our trajectories are dense.

The impact of the sampling step size W on the computation time is depicted in the top row of Figure 13. The computational cost increases significantly for very low sampling step sizes. Here, the most expensive part is writing the features to the hard disk since we have 140 times more features for $W = 2$

[17] `http://www.irisa.fr/vista/Equipe/People/Laptev/download.html`

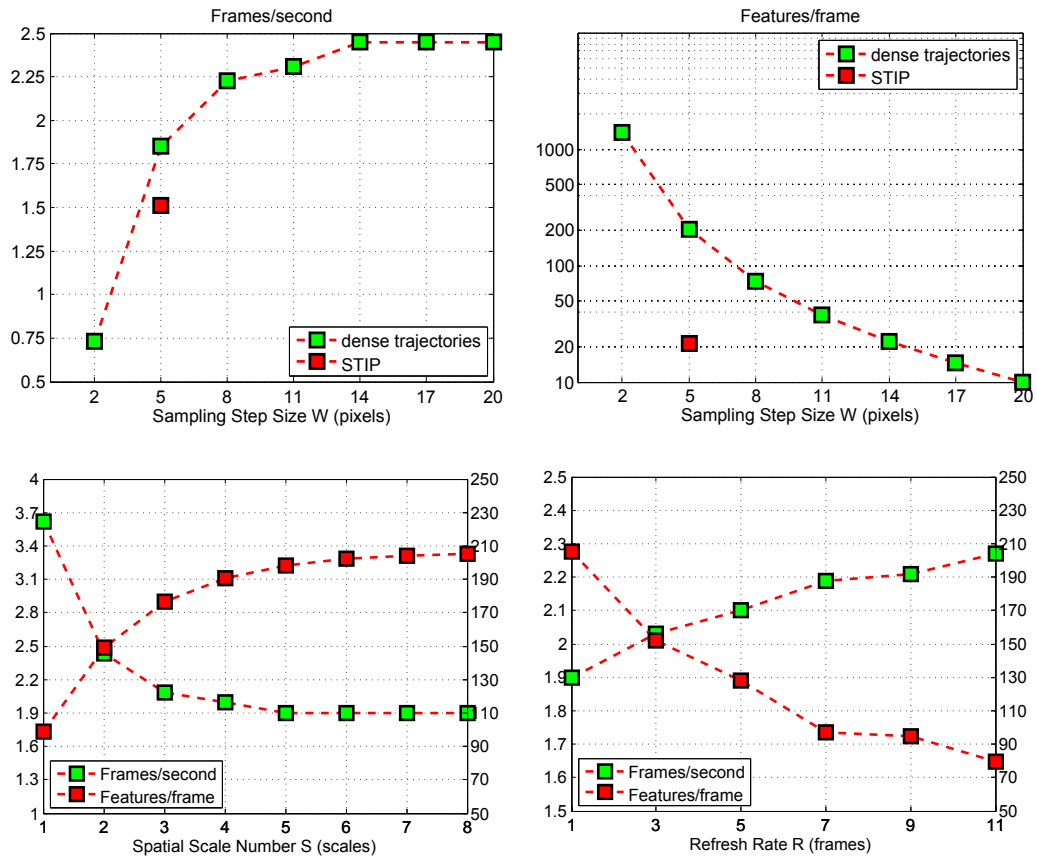


Figure 13: The top row compares the computational complexity of dense trajectories and STIP [5]. We report run-time (top, left) and the number of features per frame (top, right). The bottom row presents an analysis of the complexity w.r.t. the spatial scale number S (bottom, left) and refresh rate R (bottom, right).

compared to $W = 20$ pixels. We achieve 2.4 frames/second for $W = 14$ pixels to $W = 20$ pixels, in which case most of the time is used to compute dense optical flow fields. Using the default setting ($W = 5$ pixels), our code is slightly faster (1.8 frames/second) than the computation of STIP features (1.5 frames/second). However, we obtain about 10 times more features: 205.1 features/frame compared to 21.5 features/frame.

We illustrate the computational complexity for the spatial scale number S and refresh rate R in the bottom row of Figure 13. The speed is twice faster when features are only extracted on the original scale. The refresh rate R has only a minor impact on the computational speed. We can reduce the number of features by either decreasing S or increasing R.

6.6 Comparison to state-of-the-art results

In this section, we compare our results to the state of the art on each dataset. Table 7 displays our results with and without spatio-temporal pyramids (STP) and compares our approach to the most recent results in the literature.

KTH		YouTube		Hollywood2	
Laptev *et al*. [5]	91.8%	Liu *et al*. [45]	71.2%	Wang *et al*. [17]	47.7%
Kovashka and Grauman [53]	94.53%	Ikizler-Cinbis and Sclaroff [35]	75.21%	Taylor *et al*. [54]	46.6%
Yuan *et al*. [55]	93.7%	Brendel and Todorovic [56]	77.8%	Ullah *et al*. [43]	53.2%
Le *et al*. [57]	93.9%	Le *et al*. [57]	75.8%	Gilbert *et al*. [58]	50.9%
Gilbert *et al*. [58]	94.5%	Bhattacharya *et al*. [59]	76.5%	Le *et al*. [57]	53.3%
MBH	95.0%	MBH	80.6%	MBH	55.1%
Combined	94.2%	Combined	84.1%	Combined	58.2%
MBH+STP	**95.3%**	MBH+STP	83.0%	MBH+STP	57.6%
Combined+STP	94.4%	Combined+STP	**85.4%**	Combined+STP	**59.9%**
UCF sports		IXMAS		Olympic Sports	
Wang *et al*. [17]	85.6%	Tran and Sorokin [50]	80.22%	Niebles *et al*. [49]	72.1%
Kläser *et al*. [60]	86.7%	Junejo *et al*. [61]	79.6%	Brendel and Todorovic [62]	77.3%
Kovashka and Grauman [53]	87.27%	Wu *et al*. [63]	88.2%	Gaidon *et al*. [64]	**82.7%**
Le *et al*. [57]	86.5%				
MBH	84.2%	MBH	91.8%	MBH	71.6%
Combined	88.0%	Combined	93.5%	Combined	74.1%
MBH+STP	84.0%	MBH+STP	91.9%	MBH+STP	74.9%
Combined+STP	**89.1%**	Combined+STP	**93.6%**	Combined+STP	77.2%
UIUC		UCF50		HMDB51	
Tran and Sorokin [50]	**98.7%**	Kliper-Gross [65]	72.7%	Sadanand and Corso [66]	26.9%
MBH	97.1%	MBH	82.2%	MBH	43.2%
Combined	98.4%	Combined	84.5%	Combined	46.6%
MBH+STP	98.1%	MBH+STP	83.6%	MBH+STP	45.1%
Combined+STP	98.3%	Combined+STP	**85.6%**	Combined+STP	**48.3%**

Table 7: Comparison of the MBH descriptor and a combination of all descriptors without and with spatio-temporal pyramids (STP) to the state of the art, as reported in the cited publications.

Spatio-temporal pyramids (STP) [5] improve results on most datasets for both a single descriptor (e.g., MBH) and a combination of descriptors. With STP, the performance of MBH is improved by around 2% on YouTube, Hollywood2 and HMDB51. For the combined result, the improvement is less significant. We observe about 1% of improvement on YouTube, Hollywood2, UCF sports, UCF50 and HMDB51. The biggest improvement is obtained for Olympic Sports, where both MBH and the combined result increase by about 3% when using the spatio-temporal pyramid. There is no improvement for datasets with clean background (i.e., KTH, IXMAS, UIUC).

Table 7 also compares our approach to state-of-the-art results We can observe that it outperforms significantly the state of the art on all datasets except KTH, UIUC, and Olympic Sports where it is on par with results in the literature.

On YouTube, we significantly outperform the state of the art [56] by over 8%. On Hollywood2 “combined+STP" reports 59.9%. This is over 6% higher than the state of the art [57] that employs features learning using deep networks. On UCF Sports “combined+STP" achieves 89.1% which is around 2% better than the state of the art [53].

IXMAS is a popular multi-view action dataset. There are various experimental settings designed for cross-view action recognition. We compare to the results which are also trained and tested on all

five views. “Combined+STP" reports 93.6% and significantly outperforms the state of the art [63] by 5%. Weinland *et al*.[67] reported 93.33% using motion history volumes of 3D human body models for training and testing. These 3D exemplars were obtained based on reconstruction given the five views.

Niebles *et al*.[49] obtain 72.1% on Olympic Sports with an approach which models the temporal structure of a human action. Note that we get a similar result (i.e., 71.6%) with a single MBH descriptor. “Combined+STP" reports 77.2%, which outperforms their result by over 5%. Brendel and Todorovic [62] reported a similar result (77.3%) by designing complex spatio-temporal graphs to model the hierarchical relationships embedded in actions. Recently, Gaidon *et al*.[64] achieved 82.7% by clustering dense trajectories and modeling the relationship between the clusters via a tree structure.

On UCF50, “MBH + STP" results in 83.6%, whereas “Combined+STP" achieves 85.6%. Recently, Kliper-Gross *et al*.[65] reported 72.7% by designing descriptors to capture local changes in motion directions. On the recent HMDB51 dataset, we outperform the state of the art [66] by 16% using only “MBH". “Combined+STP" further improves the performance by 5%.

7 Conclusions

This paper introduced an approach for efficient video description based on dense trajectories and motion boundary histogram descriptors. Our dense trajectories have shown to outperform previous approaches that extract trajectories with either the Kanade-Lucas-Tomasi (KLT) tracker or by SIFT descriptor matching. Motion boundary histogram descriptors which are computed along the dense trajectories have shown to yield excellent results. They are designed to be robust to camera motion and are shown to outperform the state-of-the-art histogram of optical flow descriptor. We have evaluated our video description extensively on nine datasets and have shown that it significantly outperforms the state of the art.

Our video description is by all means not limited to a bag-of-features representation and could be applied in the context of action localization and video retrieval. The efficiency of the algorithm enables its application on large scale video data. Yet, its performance is currently limited by the quality of the optical flow available. The state-of-the-art optical flow algorithms are often computationally expensive and far from perfect, as discussed in section 6.3. Developing better optical flow algorithms suitable for large realistic video datasets is important to improve the performance of current action recognition systems.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China (NSFC) Grant 60825301, the National Basic Research Program of China (973 Program) Grant 2012CB316302, the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant XDA06030300), as well as the joint Microsoft/INRIA project and the European integrated project AXES.

References

[1] I. Laptev, “On space-time interest points,” *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[2] M. Bregonzio, S. Gong, and T. Xiang, “Recognising action as clouds of space-time interest points,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[3] 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.

[4] G. Willems, T. Tuytelaars, and L. Gool, “An efficient dense and scale-invariant spatio-temporal interest point detector,” in *European Conference on Computer Vision*, 2008.

[5] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[6] C. Schüldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *International Conference on Pattern Recognition*, 2004.

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

[8] A. Kläser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *British Machine Vision Conference*, 2008.

[9] L. Yeffet and L. Wolf, "Local trinary patterns for human action recognition," in *IEEE International Conference on Computer Vision*, 2009.

[10] P. Matikainen, M. Hebert, and R. Sukthankar, "Trajectons: Action recognition through the motion analysis of tracked features," in *ICCV Workshops on Video-Oriented Object and Event Classification*, 2009.

[11] R. Messing, C. Pal, and H. Kautz, "Activity recognition using the velocity histories of tracked keypoints," in *IEEE International Conference on Computer Vision*, 2009.

[12] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li, "Hierarchical spatio-temporal context modeling for action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[13] J. Sun, Y. Mu, S. Yan, and L.-F. Cheong, "Activity recognition using dense long-duration trajectories," in *IEEE International Conference on Multimedia and Expo*, 2010.

[14] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence*, 1981.

[15] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[16] E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in *European Conference on Computer Vision*, 2006.

[17] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *British Machine Vision Conference*, 2009.

[18] P. Sand and S. Teller, "Particle video: Long-range motion estimation using point trajectories," *International Journal of Computer Vision*, vol. 80, pp. 72–91, 2008.

[19] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *European Conference on Computer Vision*, 2010.

[20] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[21] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *European Conference on Computer Vision*, 2006.

[22] S.-F. Wong and R. Cipolla, "Extracting spatiotemporal interest points using global information," in *IEEE International Conference on Computer Vision*, 2007.

[23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[24] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *European Conference on Computer Vision*, 2006.

[25] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[26] M. Raptis and S. Soatto, "Tracklet descriptors for action modeling and video analysis," in *European Conference on Computer Vision*, 2010.

[27] W.-C. Lu, Y.-C. F. Wang, and C.-S. Chen, "Learning dense optical-flow trajectory patterns for video object extraction," in *IEEE Advanced Video and Signal Based Surveillance Conference*, 2010.

[28] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," *Image and Vision Computing*, vol. 14, pp. 609–615, 1996.

[29] N. Anjum and A. Cavallaro, "Multifeature object trajectory clustering for video analysis," *IEEE Transactions on Multimedia*, vol. 18, pp. 1555–1564, 2008.

[30] C. R. Jung, L. Hennemann, and S. R. Musse, "Event detection using trajectory clustering and 4-D histograms," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, pp. 1565–1575, 2008.

[31] A. Hervieu, P. Bouthemy, and J.-P. L. Cadre, "A statistical video content recognition method using invariant features on object trajectories," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, pp. 1533–1543, 2008.

[32] X. Wang, K. T. Ma, G.-W. Ng, and W. E. L. Grimson, "Trajectory analysis and semantic region modeling using a nonparametric Bayesian model," in *IEEE International Conference on Computer Vision*, 2008.

[33] G. Piriou, P. Bouthemy, and J.-F. Yao, "Recognition of dynamic video contents with global probabilistic models of visual motion," *IEEE Transactions on Image Processing*, vol. 15, pp. 3418–3431, 2006.

[34] H. Uemura, S. Ishikawa, and K. Mikolajczyk, "Feature tracking and motion compensation for action recognition," in *British Machine Vision Conference*, 2008.

[35] N. Ikizler-Cinbis and S. Sclaroff, "Object, scene and actions: combining multiple features for human action recognition," in *European Conference on Computer Vision*, 2010.

[36] S. Wu, O. Oreifej, and M. Shah, "Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories," in *IEEE International Conference on Computer Vision*, 2011.

[37] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[38] N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by GPU-accelerated large displacement optical flow," in *European Conference on Computer Vision*, 2010.

[39] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Proceedings of the Scandinavian Conference on Image Analysis*, 2003.

[40] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[41] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, 2011.

[42] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International Journal of Computer Vision*, vol. 73, no. 2, pp. 213–238, 2007.

[43] M. M. Ullah, S. N. Parizi, and I. Laptev, "Improving bag-of-features action recognition with non-local cues," in *British Machine Vision Conference*, 2010.

[44] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[45] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[46] M. Marszałek, I. Laptev, and C. Schmid, "Actions in context," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[47] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH a spatio-temporal maximum average correlation height filter for action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[48] D. Weinland, E. Boyer, and R. Ronfard, "Action recognition from arbitrary views using 3D exemplars," in *IEEE International Conference on Computer Vision*, 2007.

[49] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *European Conference on Computer Vision*, 2010.

[50] D. Tran and A. Sorokin, "Human activity recognition with metric learning," in *European Conference on Computer Vision*, 2008.

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

[52] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: A large video database for human motion recognition," in *IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 2556–2563.

[53] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[54] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *European Conference on Computer Vision*, 2010.

[55] J. Yuan, Z. Liu, and Y. Wu, "Discriminative video pattern search for efficient action detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 1728–1743, 2011.

[56] W. Brendel and S. Todorovic, "Activities as time series of human postures," in *European Conference on Computer Vision*, 2010.

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

[58] A. Gilbert, J. Illingworth, and R. Bowden, "Action recognition using mined hierarchical compound features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 883–897, 2011.

[59] S. Bhattacharya, R. Sukthankar, R. Jin, and M. Shah, "A probabilistic representation for efficient large scale visual recognition tasks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[60] A. Kläser, M. Marszałek, I. Laptev, and C. Schmid, "Will person detection help bag-of-features action recognition?" INRIA, Tech. Rep. RR-7373, 2010.

[61] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez, "View-independent action recognition from temporal self-similarities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 172–185, 2011.

[62] W. Brendel and S. Todorovic, "Learning spatiotemporal graphs of human activities," in *IEEE International Conference on Computer Vision*, 2011.

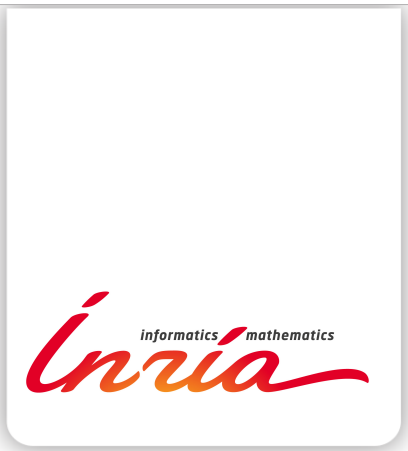
[63] X. Wu, D. Xu, L. Duan, and J. Luo, "Action recognition using context and appearance distribution features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[64] A. Gaidon, Z. Harchaoui, and C. Schmid, "Recognizing activities with cluster-trees of tracklets," in *British Machine Vision Conference*, 2012.

[65] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf, "Motion interchange patterns for action recognition in unconstrained videos," in *European Conference on Computer Vision*, 2012.

[66] S. Sadanand and J. J. Corso, "Action bank: A high-level representation of activity in video," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[67] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 249–257, 2006.



RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399