J. Vis. Commun. Image R. 41 (2016) 375–390

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

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

Spatio-temporal action localization and detection for human action recognition in big dataset ☆

Sameh Megrhi [a], Marwa Jmal [b,c,*], Wided Souidene [a,b], Azeddine Beghdadi [a]

[a] L2TI, Institut Galilée, Université Paris 13, 99, Avenue Jean-Baptiste Clement, 93430 Villetaneuse, France
[b] SERCom Laboratory, Ecole Polytechnique de Tunisie, Université de Carthage, B.P. 743, 2078 La Marsa, Tunisia
[c] Telnet Innovation Labs, Telnet Holding, Ariana, Tunisia

ARTICLE INFO

Article history:
Received 8 December 2015
Revised 4 October 2016
Accepted 28 October 2016
Available online 31 October 2016

Keywords:
Spatio-temporal action detection
Dense SURF
Optical flow
Action recognition
Selective temporal segmentation
Interest points trajectory

ABSTRACT

Human action recognition is still attracting the computer vision research community due to its various applications. However, despite the variety of methods proposed to solve this problem, some issues still need to be addressed. In this paper, we present a human action detection and recognition process on large datasets based on Interest Points trajectories. In order to detect moving humans in moving field of views, a spatio-temporal action detection is performed basing on optical flow and dense speed-up-robust-features (SURF). Then, a video description based on a fusion process that combines motion, trajectory and visual descriptors is proposed. Features within each bounding box are extracted by exploiting the bag-of-words approach. Finally, a support-vector-machine is employed to classify the detected actions. Experimental results on the complex benchmark UCF101, KTH and HMDB51 datasets reveal that the proposed technique achieves better performances compared to some of the existing state-of-the-art action recognition approaches.

1. Introduction

Action recognition is an active research field in computer vision. It represents a wide range of applications such as video surveillance, gesture interpretation, robotic vision, video search/retrieval and human-machine interaction. Recognizing human actions in videos is a challenging task due to the large intra-class variations of complex actions, poor quality and camera motion. In order to overcome these issues, a relevant video description based on dividing it into small sequences is required. In the following, we introduce our approach and provide a brief critical literature survey of the different methods developed for each component of the common video description and analysis system dedicated to human action detection and recognition.

Temporal segmentation of videos may be performed in different ways. Some methods are based on the trajectory of interest points (IP) [1,2]. Various trajectory based descriptors has been proposed in the last decades [3,4]. These descriptors are extracted either from optical flow [5,6], or by matching IP in different frames [4,5]. For such, the number of frames involved in setting the trajectory length depends on the used approach. In [4], the trajectory length is within a fixed interval while in [7] it is based on a fixed frame number in order to extract a displacement vector.

In some scenarios, action recognition is pre-processed by a motion segmentation step [8]. Thus, its performance is highly related to the segmentation algorithm. Pixel-wise techniques, namely background subtraction and temporal differencing [9], are the most straightforward methods. However, they are only effective under the consideration of static cameras. When dealing with moving cameras, these models are likely to fail as the background is continuously varying in addition to the target's motion. A recent study [10] revealed that optical flow (OF) based methods [11] are one of the most effective techniques in motion segmentation. Horn and Schunck [12] and Lucas and Kanade (L&K) [13] are the oldest yet most employed optical flow-based algorithms. Regarding their limitations toward accuracy and illumination changes, some improvements have been proposed [14]. Our method is also based on the computation of OFs of detected interest points (IP).

As for interest points detectors, SIFT (Scale Invariant Feature Transform) [15] and SURF (Speed-Up-Robust-Features) [16] descriptors are widely employed. Jurie and Triggs [17] revealed that using a regular dense grid for sampling local image patches

☆ This paper has been recommended for acceptance by Zicheng Liu.

* Corresponding author at: SERCom Laboratory, Ecole Polytechnique de Tunisie, Université de Carthage, B.P. 743, 2078 La Marsa, Tunisia.

E-mail addresses: sameh.megrhi@univ-paris13.fr (S. Megrhi), marwa.jmal@sercom-lab.com (M. Jmal), wided.mseddi@univ-paris13.fr (W. Souidene), azeddine.beghdadi@univ-paris13.fr (A. Beghdadi).

http://dx.doi.org/10.1016/j.jvcir.2016.10.016
1047-3203/© 2016 Elsevier Inc. All rights reserved.

enhances the use of interest points. Moreover, a recent evaluation of dense sampling proposed by Uijlings et al. [18] proved that dense SIFT and dense SURF descriptors may be extracted more quickly with no loss of accuracy. Dense sampling is shown to improve or produce comparable performance in different applications such as image classification [19]. In the same direction, Wang et al. [20] evaluated the use of dense sampling at regular positions in space and time for action recognition.

In practice, the complexity of dynamic scenes is considered as the biggest challenge facing motion segmentation. In fact, objects' motion is combined with both the camera and background motions. In this case, camera motion compensation is crucial. Earlier approaches to camera motion compensation relied on estimating the camera motion as a 2D affine transform or homography [21,22]. Other methods performed motion compensation at trajectory level [23]. All these works support the potential of motion compensation. However, in some cases it is almost impossible to separate the foreground and the background when there are close up captures of the human activity.

In this paper, we propose a motion segmentation algorithm based on the computation of optical flows of detected dense features. We propose to compensate the camera motion by determining the camera flow direction using the k-Nearest Neighbor (KNN) clustering algorithm and the affine motion model. Finally, humans/objects are segmented using temporal difference between two motion compensated frames. A bounding box is drawn around each detected object. Thereafter, the discriminative video segmentation is performed based on the extracted bounding boxes (BB).

To describe actions in videos, spatio-temporal (ST) local features are widely exploited [24]. ST descriptors are extracted by extending the 2D interest point to the temporal domain (1D) [25]. Willems et al. [26], proposed a method based on the extension of the Hessian matrix to the temporal domain to extract IP. Laptev et al. extended the volumetric features corner detector to extract space-time local structures [27]. Local descriptors were also extended to the temporal domain such as the histograms of oriented 3D spatio-temporal gradients [28], E-SURF [26] and the 3D-SIFT [29]. Noguchi et al., proposed a spatio-temporal SURF using Lucas-Kanade optical flow [30]. However, in [5,7], it was proven that the previous techniques suffer from inaccuracy due to the use of spatial and temporal information in a common 3D space. In fact, spatial information has different characteristics from temporal information, so associating them differently in a new scheme deserve to be more investigated and might be the cue of success of action detection in big datasets. That is to detect spatio-temporal features, various works are based on IP tracking upon a video sequence. Indeed, Sun et al., in [31], performed efficient action recognition by leveraging the motion information of trajectories. Authors in [32] proposed to describe interest point neighborhood through the distribution of the motion angles. They proposed to split optical flow components to extract the distribution of the motion trajectory orientation in the planes (t, x) and (t, y). The generated histograms describe, for every SURF based patch, its trajectory orientation angle and its displacement. Megrhi et al., in [6], proposed a method based on trajectory tracking of the SURF interest points into a frame packet. One of the latest work is proposed in [5] where descriptors based on appearance (histograms of oriented gradients), motion (histograms of optical flow) and trajectories are proposed to characterize shape (point coordinates). These approaches provided excellent performances for action recognition.

The representation of video objects as a dictionary of visual words [33,34] is, also, widely used in the task of action recognition. The distribution of the visual words is described by a histogram. The latter is then used in classification framework to separate different classes. In this paper, we exploit the BOVW method using a χ^2 Kernel Support vector machine (SVM).

The ultimate goal of this work is to introduce an efficient method to achieve accurate and fast action detection and recognition in big dataset. For action recognition, we focus on the trajectory tracking. We propose a video description based on a fusion process that combines motion, trajectory and visual descriptors. The overall proposal of the spatial-temporal segmentation and the associated architecture are shown in Fig. 1. Our work has already been partially described in [6,35]; here, we give new and more detailed explanations on the different parts of the proposed approach.

The remainder of the paper is organized as follows: The related literature is presented in Section 2. Section 3 is dedicated to the description of the motion segmentation proposed approach. Section 4 describes the selective segmentation of video sequences while the feature extraction approach is drawn in Section 5. Experiments and results are presented in Section 6. Finally, a conclusion is given in Section 7.

2. Literature review

Motion is the most important cue for studying humans actions. Thus, motion segmentation is a good way to reduce the amount of data involved in this task. However, it gets more challenging when dealing with moving cameras where the scene necessarily involves the motion of the background. At this level, camera motion compensation becomes compulsory. In the literature, some attempts have been proposed. Ikizler-Cinbis and Sclaroff [36] applied video stabilization using homography-based motion compensation approach. Nga and Yanai [37] subtracted the estimated camera flow multiplied by the camera direction from the flow of each extracted spatio-temporal keypoint. However, in this work, only camera translation in both horizontal and vertical directions is considered. Different works, such as in [21,38,22], considered 2D polynomial affine motion models to compensate camera motion. In [21,38], a model was employed to separate dominant motion, supposed to represent the camera motion, from residual motion in videos with dynamic scenes. More recently, Jain et al. [22] considered the same model. The compensated flow is computed as the difference between the original and the affine flow vectors of each point. Thereby, each vector is compensated by its own affine flow and not by the camera movement.

Video segmentation is followed by the features extraction step. In fact, to recognize human actions, different works based on local spatio-temporal (LST) features extraction have been developed [5]. Almost all existing LST descriptors are derived from the extension of a 2D spatial features or detectors to the temporal domain. Niebles et al. [39], summarized the video by space-time interest points. The Cuboids descriptor was proposed in [40], while 3D-SIFT was introduced in [29] to recognize actions in video volumes. In the same spirit [41] proposed the C2-shape features. Histogram of oriented gradient and Histogram of optical flow (HoG-HoF) based method has been presented in [34]. Authors in [26] introduced the spatio-temporal Hessian detector and the extended SURF. Other interesting works such as HOG3D [28], the local Trinary Patterns [42] and Space Time SURF [30] have been proposed.

Recently, a special focus was put on video description by tracking interest points motion [43]. This allows exploring several motion cues such as velocity [44,45], orientation [46,47], location [48], trajectory curves [49], trajectory parts [3] or different motion cues combinations [6]. Moreover, Sun et al. [4], encode the SIFT trajectory to extract spatio-temporal context models. Trajectory patterns can be extracted using a tracker such as the KLT (Kanade-Lucas-Tomasi) tracker [13] which is commonly employed

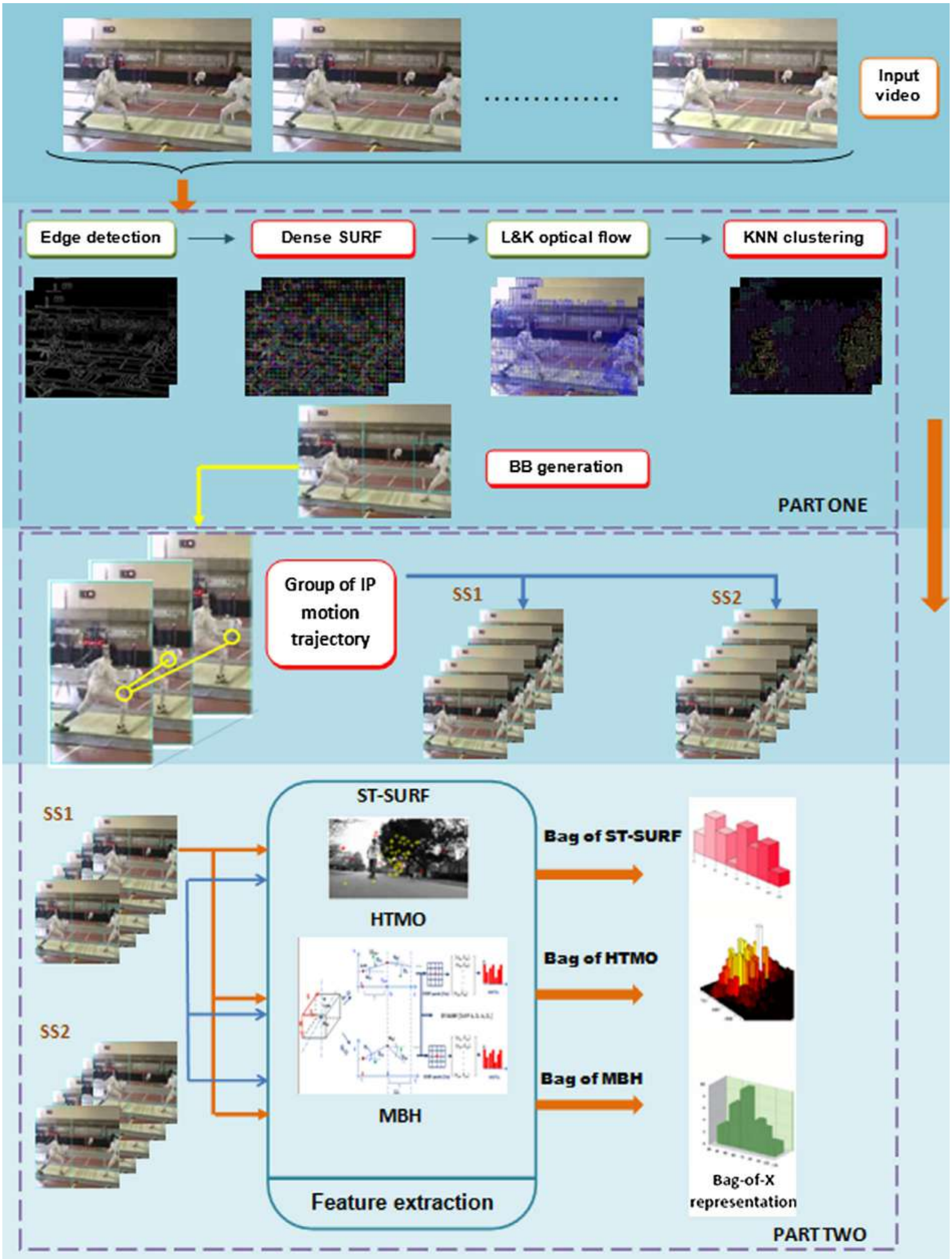

Fig. 1. Overview of the proposed approach. The framework is composed from two parts: motion detection and segmentation (PART ONE) and action recognition (PART TWO). The shot is taken from the UCF101 dataset for "Fencing" category.

in videos [50]. Authors in [31] used both SIFT and KLT features to extract long duration trajectories. Wang et al. proposed dense trajectory tracking to encode temporal information [7]. They suggested to use dense optical flow to densely track detected interest points [5]. They proved that trajectory tracking is an intuitive and successful approach in several public datasets.

Trajectory segmentation is another critical task for trajectories description. To segment the trajectory, some methods based on trajectory clustering [51,52] and moving object trajectory tracking have been proposed. In order to detect "phoning" and "standing-up" actions, authors in [53] used a sliding window classifier to extract temporal information and a human tracking process to extract trajectory information.

More recently, in [5], a new scheme is proposed to characterize dense trajectories in order to preserve trajectory smoothness. The trajectory attributes are then extracted by concatenating interest points trajectory in successive frames with a limited length of 15 frames. Finally, a trajectory shape descriptor which characterizes the displacement is computed.

In our work, we proceed clearly differently from the previous works. In the proposed approach, the frames length are set based on a simple yet efficient, automatic selective snippets segmentation to define actionlets temporal extents in order to avoid manual labeling as done in [54,53]. This allows to work only on snippets containing significant motion and allows also to reduce the cost of computing trajectory shape descriptors. Furthermore, the trajectory orientation is exploited to ensure relevant description of motion and displacement of moving objects/humans. The spatio-temporal feature coordinates describe the location of the trajectory, and henceforth, captures space-time attributes of an actionlet.

Another challenging issue is to ensure robustness of extracted features to camera motion and varying background. The insight behind the success of several proposed video descriptors is the use of static camera and uniform background [55,56]. Although, many schemes have been proposed to reduce camera motion [57,22], this problem remains unsolved in some cases. It is the purpose of this work to develop a video presentation which discards camera motion without sacrificing significant human action cues.

To this end, the motion boundaries histogram descriptor (MBH), derived from the optical flow gradient, is used as done in [58]. It removes constant motion and preserves significant one. MBH was employed in various action recognition schemes [5,7]. It provides more interesting results when applied to video containing important camera motion. MBH is certainly not dedicated to remove camera motion, but combined with the spatio-temporal SURF (ST-SURF) proposed by [6], it contributes significantly to compensate camera motion.

Descriptors extraction step is followed by a classification task based on code-book generation. Many approaches were proposed to extract a code-book for action recognition. A code-book can be generated using various techniques including, but not limited to, Random forest [59,60], Sparse code-book learning [61,62] or bag of visual words (BOVW) [33,34,5]. The BOVW approach achieved good results in action recognition in both image [63] and video analysis [64]. This is owing to the orderless feature presentation of BOVW that discards features spatial position and inter-relationship between the extracted visual words. However, the accuracy of BOVW decreases when the size of the database is huge in the case of more realistic scenes with many actors and rich background. Therefore, to incorporate spatial information, spatio-temporal pyramid is a relevant choice [34,65,5]. This approach has been introduced for analyzing and recognizing natural scenes categories [66]. The basic idea is to divide the image into increasingly size sub-regions then extract histograms of local features detected inside each sub-region.

To overcome these problems, we propose, for compensating the camera motion, to first determine the direction and magnitude of the dominant motion using a clustering of optical flow vectors, then applying the affine motion model. Once this step is achieved, we may process as if the camera is static.

In our work, spatial information is injected in the video description by a pattern called Motion Distance (MD). Consequently, there is no need for extra computation to add spatial information into the BOVW approach.

3. The proposed human motion detection and segmentation scheme

In order to reduce the amount of data involved in the task of action recognition, the proposed approach aims to detect and segment moving objects in a moving field of view. To reach this goal, interest points are first densely detected and extracted with a temporal step of N frames. Second, optical flows of detected keypoints between two frames are computed by the iterative Lucas & Kanade optical flow using a pyramidal representation [14]. Then, the resulting vector field is submitted to a flow clustering process which splits the list of flow vectors into clusters having similar flow direction and different from the direction of vectors in other clusters. Based on the clustering results, camera motion direction is determined and compensated in order to extract foreground features.

3.1. Computation of optical flow

In a given image, some parts, such as the sky or the roof, have almost the same color distribution. These parts do not, generally, contribute with useful information and add noise in the estimated optical flow. In order to overcome this drawback while preserving the most important structural features, image edges are first detected using the canny edge detector [67]. As follows, all steps of the motion segmentation process will be applied on the edge frame. Once the set of interest points densely extracted from the edge frame is defined, we track them over the next edge frame using the iterative Lucas & Kanade (LK) optical flow using a multi-resolution scheme. Fig. 2 draws an example of LK optical flow before and after edge detection. It is clear that employing the edge detection phase leads to discard various erroneous OF vectors. Optical flow computation results in a set of four-dimensional vectors V:

$$V = \{V_1 \cdots V_N | V_i = (x_i, y_i, a_i, m_i)\} \tag{1}$$

where x_i and y_i are the image coordinates of keypoint i; a_i and m_i are respectively the motion direction and magnitude of i. Note that a_i (respectively m_i) corresponds to the direction of the vector (respectively the distance) from (respectively between) keypoint i in frame t and its corresponding feature in the next frame. Generally, optical flow is computed between two successive frames. However, the result may be unstable when objects either move too fast, too slowly or stop between successive frames. In this paper, we propose to extract keypoints and compute optical flow with a temporal step size of N frames. In fact, the choice of the temporal step value varies according to the type of the video. For example, in sports videos such as running or swimming, motion is large-scale. In this case, in order to obtain more information about the motion, it is better to choose small value of N. On the other hand, in videos of everyday activities such as talking or writing, motion is rather small. In such videos, N could be chosen to be large.

The computation of optical flow vectors also allows the removal of static features that correspond to pixels with optical flow component magnitudes lower than a threshold T in both the x and y directions. Based on several observations, we empirically set the minimum motion magnitude to 0.5 pixel per frame.

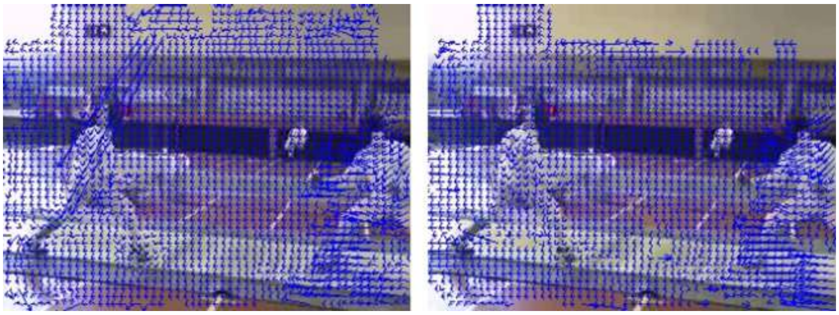

Fig. 2. Results of LK optical flow computation before (left) and after (right) edge detection.

3.2. Detection and compensation of camera motion

Once the number of extracted dense keypoints and their associated motion vectors are obtained, we separate local motions belonging to moving objects from the camera motion. In this part of the study, the problem of camera motion compensation is solved by checking the existence of camera motion based on motion vectors. If camera motion is detected, the direction and magnitude of this camera motion are determined before moving to the next step. Then, camera motion is compensated by applying affine transformations to the original frame.

Camera motion detection: In order to find out how the camera moves at each frame, our approach is based on the assumption that if most points shift to the same direction, camera motion exists and has the same direction as the moving points. The detection of camera motion is derived from analyzing optical flows between two frames in a frame set. Therefore, a clustering of optical flow vectors is proposed in order to eliminate outliers and determine the direction of camera motion. In view of our real-time requirements, it is desirable to have a low number of clusters with similar optical flow vectors. It's worth to notice here that we do not seek to group motion vectors having the same magnitude or deviation. We are interested only on the direction of the motion. Fig. 3 depicts the eight possible directions of the camera motion: six in the horizontal direction: forward (up, down or right) or backward (up, down or left), and two in the vertical direction (up or down). In order to segment flow field into different groups, the k-Nearest Neighbor (KNN) clustering algorithm is employed.

After performing the KNN clustering, several small clusters may appear. These clusters do not belong to a dominant cluster and are not relevant to the purposes of the study. Therefore, clusters with a size lower than a certain threshold are discarded. Fig. 4 presents examples of optical flow clustering using KNN. Each of the eight directions of the camera is represented by a different color. In these representations, it is easy to distinguish the moving objects from the background as well as determining the direction of camera motion. In the first three images, the camera is moving in a different direction than the humans. In the last one, the man on the left has the same motion direction as the camera (presented in the same color), but their velocities are different. Therefore, the camera motion compensation should be performed.

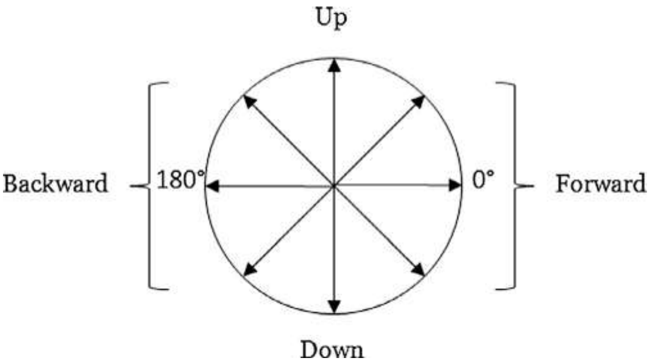

Fig. 3. Possible directions of camera motion.

Since we assumed that camera motion exists if most points move in the same direction, we determine the size of each of the eight clusters. We, then, compare the size of the largest cluster to the minimal required proportion of moving points set to $\frac{N}{2}$ where N is the total number of detected points. Therefore, camera motion exists if Eq. (2) is satisfied:

$$\sup_{i\in 1,\dots,8}\{s_i\} \geqslant \frac{N}{2} \tag{2}$$

where s_i is the size of cluster i and i is the number of the cluster. As an example, in the first row of Fig. 4, we can easily interpret that purple is the dominant color. Hence, the camera is in motion and it is moving horizontally to the left. If the above condition is not satisfied, then the camera is supposed to be in rest and if it is detected as being in motion, then the camera motion magnitude and deviation are computed using on the following equations:

$$m_m = mean|f_i| \tag{3}$$

$$\theta_m = mean(\theta_{f_i}) \tag{4}$$

Here, f_i and θ_{f_i} refer, respectively, to the flow and deviation of point i. m_m and θ_m refer, respectively, to the camera flow magnitude and deviation.

Camera motion compensation: In videos captured by a handheld camera, camera motion is random. This motion is a combination of translation and rotation. In Nga et al.'s work [37], only the camera translation is considered. Camera motion is compensated by subtracting the camera flow from the original flow of each SURF keypoint. As a result, the camera motion will not be correctly compensated if the motion is, for example, oblique. We propose to solve this problem by applying affine transformation to each frame in which camera motion is detected. The affine model [68] incorporates transformation such as translation, rotation, and scaling (compressions or expansions). The transformation can be described as:

$$I' = D \times I + d \times T \tag{5}$$

where I is the original frame; I' is the transformed frame; $D = \begin{bmatrix} s_x d_{xx} & s_y d_{xy} \\ s_x d_{yx} & s_y d_{yy} \end{bmatrix}$ is the deformation matrix accounting for rotation and scaling; $d_{xx}, d_{xy}, d_{yx}, d_{yy}$ are the rotation parameters and s_x

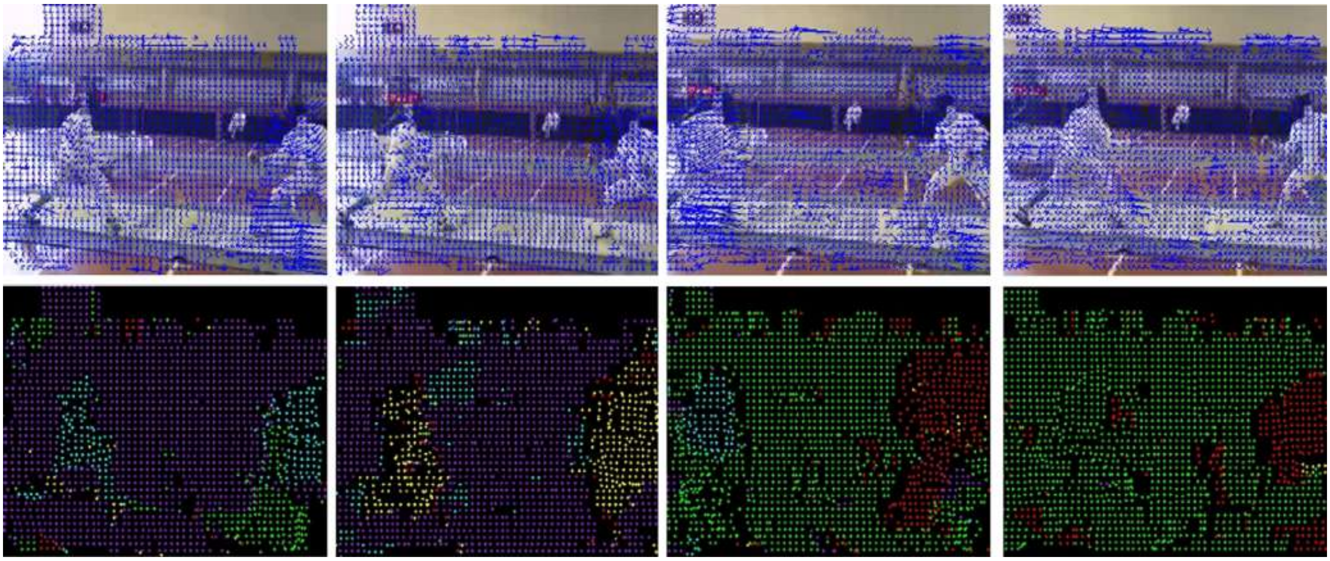

Fig. 4. Optical flow clustering using KNN algorithm: the first row presents optical flows between two frames taken from a video sequence while the second row displays the results of KNN clustering. The keypoints are grouped into eight clusters with different colors depending on the flow direction. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and s_y are the scaling ratios in the x and y directions. $T = \begin{bmatrix} d_x \\ d_y \end{bmatrix}$ is the translation vector. d equals 1 if the camera moves in a positive direction or -1 if the camera moves in a negative direction.

In this work, we take under consideration only translation and rotation motions. The parameters s_x and s_y from the deformation matrix are equal to 1. Hence, I' from (5) becomes:

$$I' = \begin{bmatrix} \cos\theta_m & -\sin\theta_m \\ \sin\theta_m & \cos\theta_m \end{bmatrix} \times I + \begin{bmatrix} m_{mH} \\ m_{mV} \end{bmatrix} \tag{6}$$

Here m_{mH} (respectively m_{mV}) refers to the camera flow magnitude when the camera translates horizontally (respectively vertically). In case of horizontal motion, $m_{mV} = 0$ and in case of vertical motion, $m_{mH} = 0$. Unlike in [22,37] where the motion of each flow vector is compensated for independently, in our work, we apply the affine model on the whole image.

3.3. Motion segmentation

After compensating the camera motion, we reach a situation similar to when the camera is static. Here, moving objects are segmented using a pixel-wise technique known as temporal difference. It is the simplest method for extracting moving objects and is robust in dynamic environments. It is similar to background subtraction techniques. The only difference between them is that the background model for temporal difference is the previous frame.

This algorithm classifies a new pixel as being a foreground pixel whenever $\|I(x,y) - I_{prev}(x,y)\| \geqslant T_h$ where T_h is a user defined threshold. In this work, T_h is set experimentally to 100. The output is a binary image. However, due to camera noise and limitations of the background model, the foreground mask typically contains numerous small "noise" clusters. These erroneous clusters can be removed by applying a noise filtering algorithm to the foreground mask. Removing them at an early stage is desirable since they can interfere with later post-processing steps.

In general, morphological operations are performed to remove noise and extract significant information from images. In our system, we used both morphological erosion and dilatation, using a structuring element with size 2 × 2, to remove noise and unwanted objects. After that, objects, including many small holes and separated pixels, are connected into one cluster using the dilatation operation. Small and useless clusters are removed by setting limitation on their sizes. The remaining clusters represent the moving objects.

Finally, a bounding box is drawn around each detected object. The aforementioned steps of our proposed method for motion segmentation are applied to an input video with a temporal step of size N. Thus, the detected objects in the remaining frames (the frames between frame (n) and frame $(n+N)$) need to be tracked. To accomplish this, we employ a template matching technique known as normalized cross correlation [69]. Fig. 5 emphasizes the effectiveness of our motion segmentation method. It can be observed that almost only local motions remain which are then employed, after filtering noise, to segment the motion. Our method succeeded to eliminate the motion induced by the camera and thus keeping only the motion of humans/objects. However, in some cases, the process of camera motion compensation may have a reverse effect on motion segmentation. In fact, in some frames, two or more dominant planes coexist. Hence, the camera motion direction and deviation will not be determined correctly. For example in the fourth row of Fig. 5, the motion of two players can be easily detected before camera motion compensation. When applied, the latter adds some noise to the frame. At the end, we were able to solve this problem using morphological operations.

In the case where no camera motion is detected, we admit that the detected flow belongs to the objects/humans in motion. Hence, instead of applying, as we did previously, the temporal differencing technique, here, we propose to apply a second clustering of optical flow vectors based on the degree of similarity of their magnitudes, angles and closeness, under the assumption that optical flows of a single person/object have similar characteristics.

We assume that two optical flow vectors, f_i and f_j, belong to the same cluster if the following assumptions are satisfied:

$$|l_i - l_j| \leqslant l_{th} \tag{7}$$

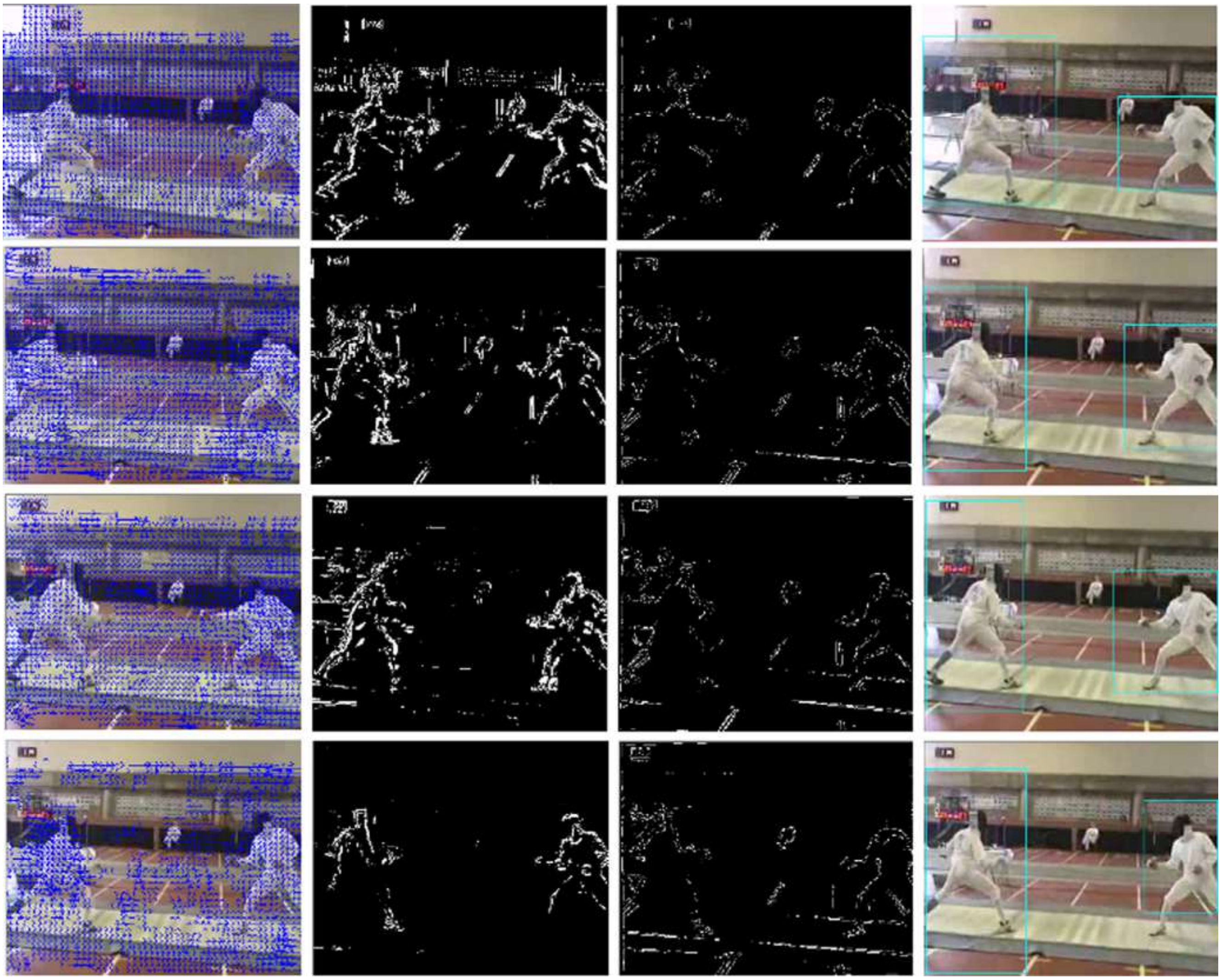

Fig. 5. Results of our proposed method for motion segmentation. Camera motion exists in all the sequence. The first column presents a frame set of consecutive frames containing camera motion on which optical flow is drawn. The second column refers to the motion segmentation results before camera motion compensation. The third column shows the results of motion segmentation after camera motion compensation. Finally, the last column is the final segmentation after applying morphological operations.

$$|\theta_i - \theta_j| \leqslant \theta_{th} \tag{8}$$

$$|posX_i - posX_j| \leqslant posX_{th} \tag{9}$$

$$|posY_i - posY_j| \leqslant posY_{th} \tag{10}$$

where l_i and l_j are the magnitudes of f_i and f_j. θ_i and θ_j are the deviations (angles) of f_i and f_j. (X_i, Y_i) and (X_j, Y_j) are the coordinates of optical flow vectors. Finally, $l_{th}, \theta_{th}, posX_{th}$ and $posY_{th}$ are the thresholds for optical flow clustering.

All detected flow vectors are compared two-by-two based on these similarity comparisons leading to form a fixed number of clusters. Noisy and meaningless clusters, are removed. The remaining clusters belong to the foreground. A bounding box is drawn around each one.

Fig. 6 presents the segmentation results derived from the optical flow clustering technique as well as the results of using the frame difference technique. The OF clustering technique (row 5) achieves better segmentation results. It succeeds to capture the whole human motion, whereas the second technique (rows 3 and 4) leads to loss of information and only some parts of the motion are segmented.

4. Selective snippets (SS) and Group of SURF (G-SURF) segmentation

One of the main objectives of the proposed method is to reduce computational time. This could be achieved by reducing the number of the video frames to be analyzed. For this purpose, we propose the use of concepts of selective snippets and the group of SURF (G-SURF). A selective snippet is a video portion that contain action. Considering three successive frames $(n, n+1, n+2)$, a detected SURF in frame n can be detected in the same location in the following frame $(n+1)$. Also it can simply disappears or be detected in another spatial location if the SURF moves. Therefore, a trajectory description to follow the motion of this point can be extracted. Considering α, as shown in Fig. 7, the angle between the lines segments supporting the motion of a SURF from the couple of frames $(n, n+1)$ and $(n+1, n+2)$, we compare α to α_{max} (α_{max} is empirically set) to segment a succession of frames (SS) in

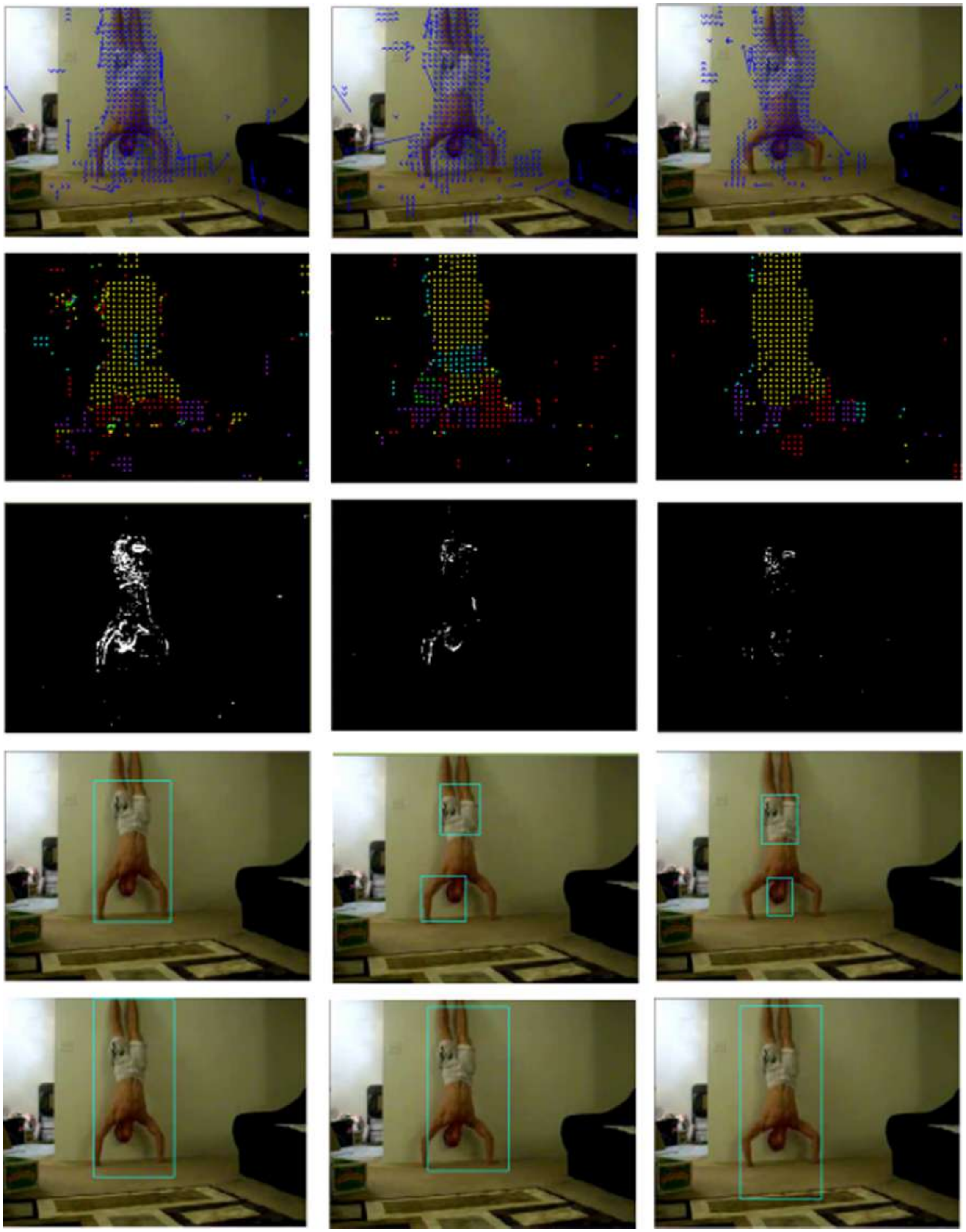

Fig. 6. Results of motion segmentation in videos acquired by static camera. The first row presents a set of consecutive frames on which optical flow is drawn. The second row refers to optical flow clustering using KNN clustering. The third and forth rows show the results of temporal differencing technique. Finally, the last row is the result of motion segmentation after optical flow second clustering.

which each SURF has an α lower than α_{max}. Let $D_{n,n+1}$ be the displacement vector of a given SURF from the frame (n) to the frame $(n+1)$; $D_{n,n+2}$ from the frame (n) to the frame $(n+2)$.

$$D_{n,n+1} = (Dx_{n,n+1}, Dy_{n,n+1}, Dt_{n,n+1}) \tag{11}$$

and

$$D_{n+1,n+2} = (Dx_{n+1,n+2}, Dy_{n+1,n+2}, Dt_{n+1,n+2}) \tag{12}$$

$$\alpha = \arccos \frac{D_{n,n+1} \cdot D_{n+1,n+2}}{\|D_{n,n+1}\| \times \|D_{n,n+2}\|} \tag{13}$$

Note that, within a SS, all SURF motions are lower than α_{max}. In order to avoid an over-sized SS, we introduce the concept of G-SURF. This is a parameter defining the number of grouped SURF empirically tuned. The grouping technique is then performed over successive SURF detected in a reference frame. By defining G-SURF, an average motion angle (α_{avg}) is computed and compared to α_{max}.

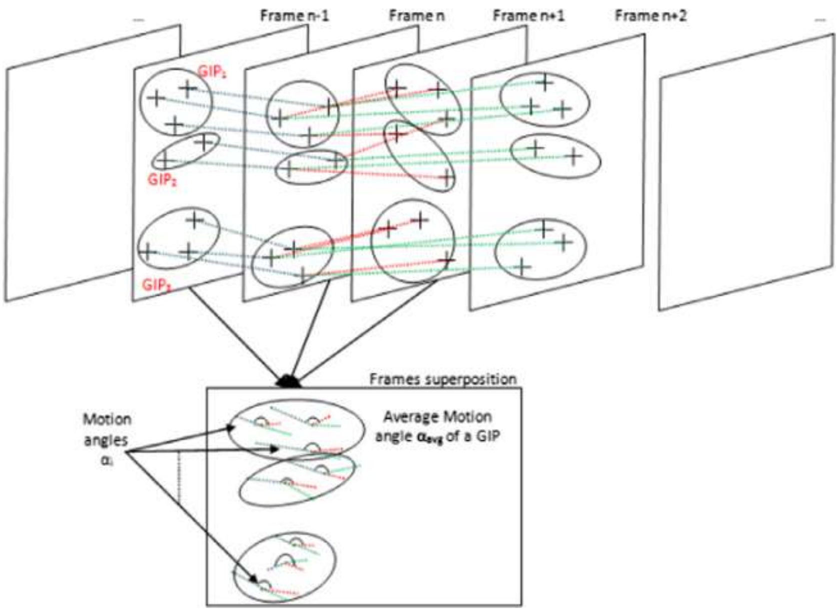

Fig. 7. IPs trajectory tracking for FPs segmentation.

The more SURFs are, the less α_{avg} is sensitive to motion and the more the SS will have extended borders. The extracted keyframes are called t_{min} and t_{max}. The main steps of the proposed segmentation algorithm are given below (Algorithm 1).

Algorithm 1. Proposed algorithm for FPs segmentation.

```
Require: $I$ - input video
  $\alpha_{min}, \alpha_{max}$ - motion angles
Ensure: $f, t_{min}, t_{max}$
  Step1: IP extraction from frames $\{f_n, f_{n+1}\}$;
  Step2: Groups of IPs defined;
  Step3: Compute the line supporting the motion;
  Apply the above three steps to $\{f_{n+1}, f_{n+2}\}$;
  Compute the angle between each motion line;
  Extract $\alpha_{avg}$ for each GIP;
  if $\alpha_{avg} \leqslant \alpha_{min}$ then
    go to the next frame;
  else
    Compare $\alpha_{avg}$ to $\alpha_{max}$;
  end if
  repeat
    previous steps
  until $\alpha_{avg} \geqslant \alpha_{max}$
```

5. Feature extraction

Action recognition is a challenging computer vision task. As mentioned in the introduction, several descriptors have been proposed to achieve high quality action detection. In this section, we describe in details the main stages of the used descriptors in our process.

5.1. Local interest points extraction

In 2004, Lowe [16], presented an interest point called scale invariant feature transform (SIFT). SIFT strikes a balance between robustness and fast computational time against image scale and rotation. SIFT descriptor was successfully used in image recognition and retrieval [16]. However, SIFT extraction process is very slow. Bay et al. [16] proposed a speeded-up version of SIFT called SURF. The SURF detection process is based on the determinant of the Hessian matrix (HM). In fact, HM is not only fast and accurate, but it also allows to extract both scale and location cues [16]. For a given $IP = (x, y)$ located in a frame f, the HM located at IP with the scale σ is defined as

$$H(IP,\sigma) = \begin{pmatrix} L_{xx}(IP,\sigma) & L_{xy}(IP,\sigma) \\ L_{xy}(IP,\sigma) & L_{yy}(IP,\sigma) \end{pmatrix} \qquad (14)$$

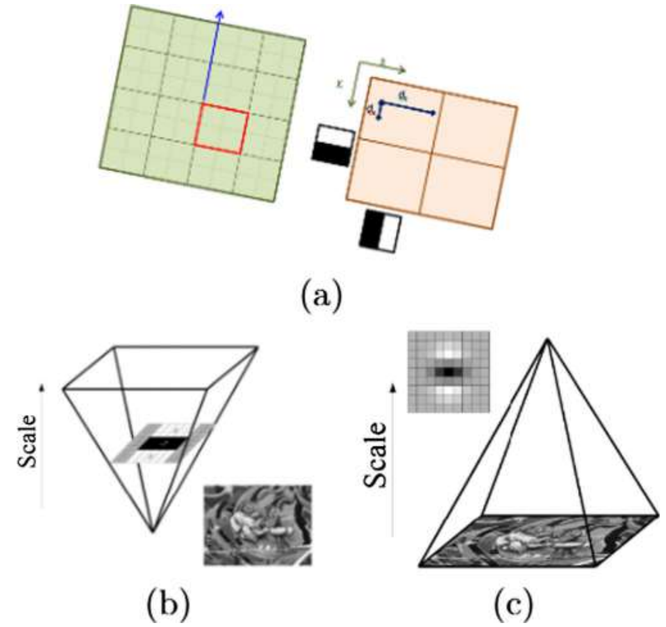

Fig. 8. A general outline of SIFT SURF extraction. (a) Box filter to estimate SURF features; (b) scales space in the SURF extraction; (c) scale space in the SIFT extraction.

where $L_{xx}(IP,\sigma)$ is the result of the convolution of the frame f in IP with the Gaussian second order derivative $\frac{\partial^2 g(\sigma)}{\partial x^2}$. This filter is approximated by using box filter (see Fig. 8(a)). Henceforth, the determinant of the approximated HM becomes:

$$det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \qquad (15)$$

Instead of using scale space representation based on pyramid decomposition, in [16], the filter size is up-scaled while keeping the original image size (see Fig. 8). To accelerate the computation time, authors employ integral images. Then, they define a square region centered by the detected interest points and characterized by a reproducible orientation. This region is then splitted into 4×4 sub-regions. A four Haar wavelet responses are extracted from every sub-region. Finally, the 64D SURF descriptor is extracted. The experiments turned out that SURF is three times faster than SIFT with a reasonable accuracy. In this work, the feature extraction solution given in [16] is adopted to extract IP. This choice is motivated by the speed up of the extraction step, robustness, the smaller size of this feature and its reliable performances attested in several datasets for action recognition [30].

5.2. Motion trajectory extraction

The motion trajectory detection tracking and extraction are based on the following steps:

5.2.1. Optical flow computation

Features tracking is performed by estimating optical flow. To increase optical flow estimation accuracy, several methods derived from the Horn and Schunck (HS) Optical flow formulation [70] have been proposed. Sun et al. [70] proposed an algorithm to approximate an optimized computationally tractable objective function, based on the original HS formulation. First, a median filtering is used to denoise the flow field. The pre-filtering of the frames reduces the influence of illumination changes. By exploiting relation between median filtering and L1-based denoising, it has been proved that algorithm relying on a median filtering step allows to optimize a different objective that regularizes the flow over a large spatial neighborhood [70]. It is filtered using a bilateral weight that depends on the spatial and the color value distance of the pixels as done in bilateral filer. The resulting algorithm ranks 1st in both angular and end-point errors in the Middlebury evaluation [70]. The initially computed optical flow serves in many blocks in the proposed framework. This reduces feature extraction computational time.

5.2.2. Trajectory tracking

Every selective snippet corresponds to a volume of frames in the 3D space called SS Volume (SS_v). This cubic volume is characterized by:

- The frame number (FN) varying from 1 to t_{max}.
- The frame surfaces dimensions (FS) varying from x to x_{max} in the x direction, and from y to y_{max} in the y direction.
- The SS cubic volume center (SS_{cc}) coordinates.

A given interest point $IP = (x,y,t)$ is defined by its spatial position (x,y) and its temporal cue t. In frame $(t+n)$, the IP undergoes a displacement u in the x direction, and v in the y direction defined as, $IP(t+n) = (x+u, y+v, t+n)$. In all our experiments, unless mentioned otherwise, we consider only moving interest points when $u \neq 0, v \neq 0$. In every pre-defined SS_v, the 3D direction (u, v, n) is the direction of the IP motion. The motion vector is calculated by the Sun et al. [70] optical flow approach. Our main contribution consists on the use of motion trajectory orientation to describe IP displacement, instead of using directly the optical flow fields (u, v, n). In-fact, the motion vector in the 3D space can be found by the intersection of two orthogonal planes to the plane (t,x) and the plane (t,y). To extract IP motion trajectory orientation, we project its motion vectors onto the planes (t,x) and (t,y) of the SS_v to define an angle for each projection of the first angle $\propto_x$ between optical flow and the plane (t,x), the angle $\propto_y$ between the plane (t,y) and the motion vector. Fig. 9 illustrates the cube and its projection into the planes (t,x) and (t,y).

$$\propto_x = 90 - \frac{180}{\Pi}\arctan\left(\frac{u}{n}\right), \propto_y = 90 - \frac{180}{\Pi}\arctan\left(\frac{v}{n}\right). \qquad (16)$$

The projection of each $SURF's$ motion vector on the planes (t,x) and (t,y) yields to two lines L_x and L_y. The orthogonal projection of SS_{ccx} and SS_{ccy} onto the lines L_x and L_y allows computing the two distances D_x and D_y between the SS_v center and the lines supporting the motion vectors (L_x and L_y).

For an IP located at (x,y,t), the distances D_x and D_y are given by:

$$D_x = D_{xu} - D_{tv}, D_y = D_{yv} - D_{tu} \qquad (17)$$

where

$$D_{xu} = (x - x_{max}/2)cos\left(180/\Pi arctan\left(\frac{u}{n}\right)\right) \qquad (18)$$

$$D_{tv} = (t - t_{max}/2)sin\left(180/\Pi arctan\left(\frac{v}{n}\right)\right) \qquad (19)$$

$$D_{yv} = (y - y_{max}/2)cos\left(180/\Pi arctan\left(\frac{v}{n}\right)\right) \qquad (20)$$

$$D_{tu} = (t - t_{max}/2)sin\left(180/\Pi arctan\left(\frac{u}{n}\right)\right) \qquad (21)$$

where t_{max}, x_{max} and y_{max} are the dimensions of the SS volume with t_{max} depending on the number of the frames contained within a segmented (SS_v). In the following, D_x and D_y describe the motion trajectory location in the 3D volume generated from the successive frames.

5.2.3. Histogram of motion trajectory orientation (HMTO)

A wide range of histograms have been proposed in the literature for action recognition description. Some of them focus on extracting motion cues such as [34] or MBH [58]. While other extract spatial information i.e., HOG descriptor [25]. In this paper, we introduce a novel descriptor called motion trajectory orientation histogram (HMTO). The most valuable property of this descriptor is that it is splitted in order to capture motion trajectory orientation patterns in both (x,t) and (y,t) directions. To gain more accuracy, we extract both $HMTO_x$ and $HMTO_y$ from a SURF centered patch. The patch is a square region with size $20s$ where s represent the current scale. Furthermore, for every pixel in the detected patch, we compute the optical flow. Then, we extract the direction parameters α_x and α_y. These are considered as the angular votes in $HMTO_x$ and $HMTO_y$. To use the trajectory cues to track actions, we propose to bin them based on the absolute motion distance. Finally we extract 8 bins histogram $HMTO_x$ and $HMTO_y$. These histograms are finally L_2 normalized (see Fig. 10).

5.2.4. Motion boundary histogram (MBH)

The motion boundary histogram (MBH) was introduced in [58] to detect actions. MBH contains the distribution of the gradient of the optical flow fields in both x and in y directions. Hence, it captures salient optical flow changes while suppressing static motion usually derived from camera motion. The final MBH_x and MBH_y are 96D $(2 \times 2 \times 3 \times 8)$ features set. In this work, we used MBH, not only for its aptitude of reducing camera motion, but also as a

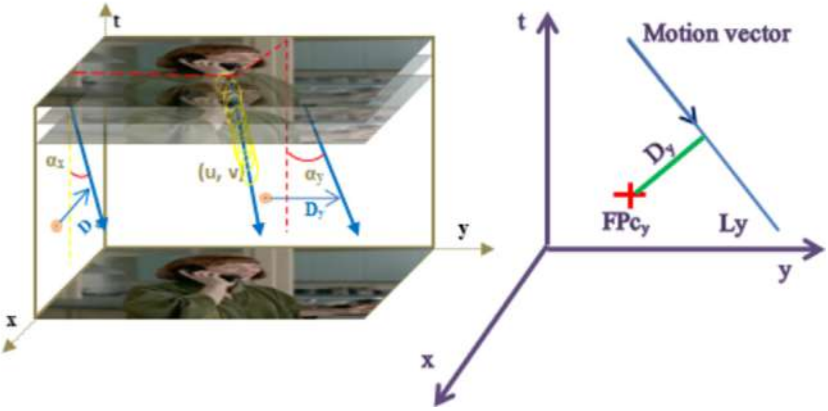

Fig. 9. The projection of a motion vector in the adjacent planes.

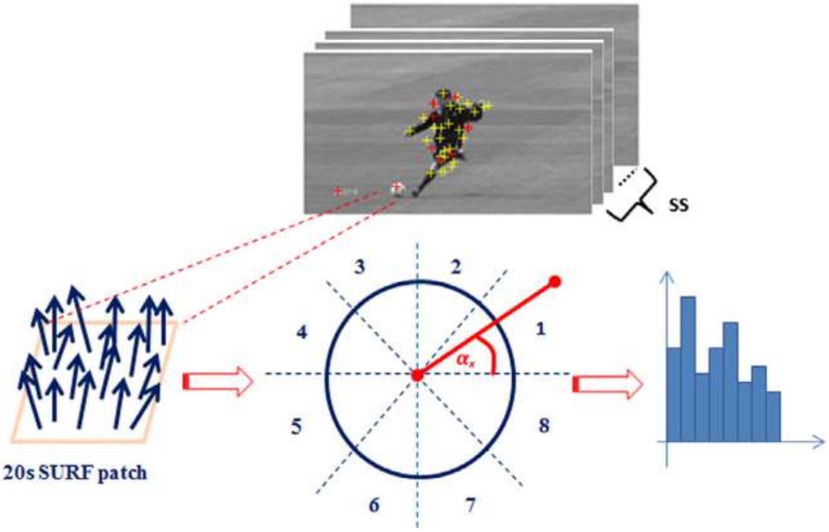

Fig. 10. Overview of HMTO extraction.

motion descriptor for its action recognition discriminative power attested in the state-of-the-art [58,5].

5.2.5. Spatio-temporal SURF (ST-SURF)

ST-SURF was introduced by [6]. The main idea is to detect the trajectory of a SURF point by tracking its motion trajectory. The authors use Hessian Matrix to detect salient points. Then, they extract all SURFs in a given video. Finally they compute a 68D spatio-temporal SURF called ST-SURF. The results given by their proposed approach are encouraging but still below the state-of-the-art. In this paper, we give an optimized ST-SURF extracted over a SS. This step is based on a dense SURF extraction, which boosts the information detection step. We combine ST-SURF with other descriptors to capture maximum spatial and temporal cues. We choose ST-SURF for many reasons. First, it contains spatial information driven by the SURF and temporal information driven by the optical flow, the size of this descriptor and finally it provides localization information. The latter will add spatial information to the bag of words encoding step.

6. Experiments and results

A highlight of the experimental settings and results are presented in this section.

6.1. Dataset

Our proposed algorithm is evaluated on three complex benchmarks for action recognition: UCF101, KTH and HMDB51. Examples from each dataset is presented in Fig. 11.

UCF101 [71]. UCF101 is one of the largest realistic datasets for action recognition. It includes a total number of 101 action classes which are divided into five categories: Human-Object Interaction, Body-Motion, Human-Human Interaction, Playing Musical Instruments, Sports.

Clip Groups: The clips of one action class are divided into 25 groups which contain 4–7 clips each. The clips in one group share some common features, such as the background or actors. The videos are downloaded from YouTube [72] and the irrelevant ones are manually removed. All clips have fixed frame rate and resolution of 25 FPS and 320 × 240 respectively.

KTH [73]. The KTH dataset is commonly used as a public benchmark test of spatio-temporal features. It contains six kinds of actions: walking, running, jogging, boxing, hand waving and hand clapping. We consider six action classes performed by twenty-five persons in four different scenarios (indoor, outdoor, different clothes outdoors, scale outdoors) with a total of 2391 video samples, all with a homogeneous and static background. The average length of videos in the KTH dataset is about 20 s.

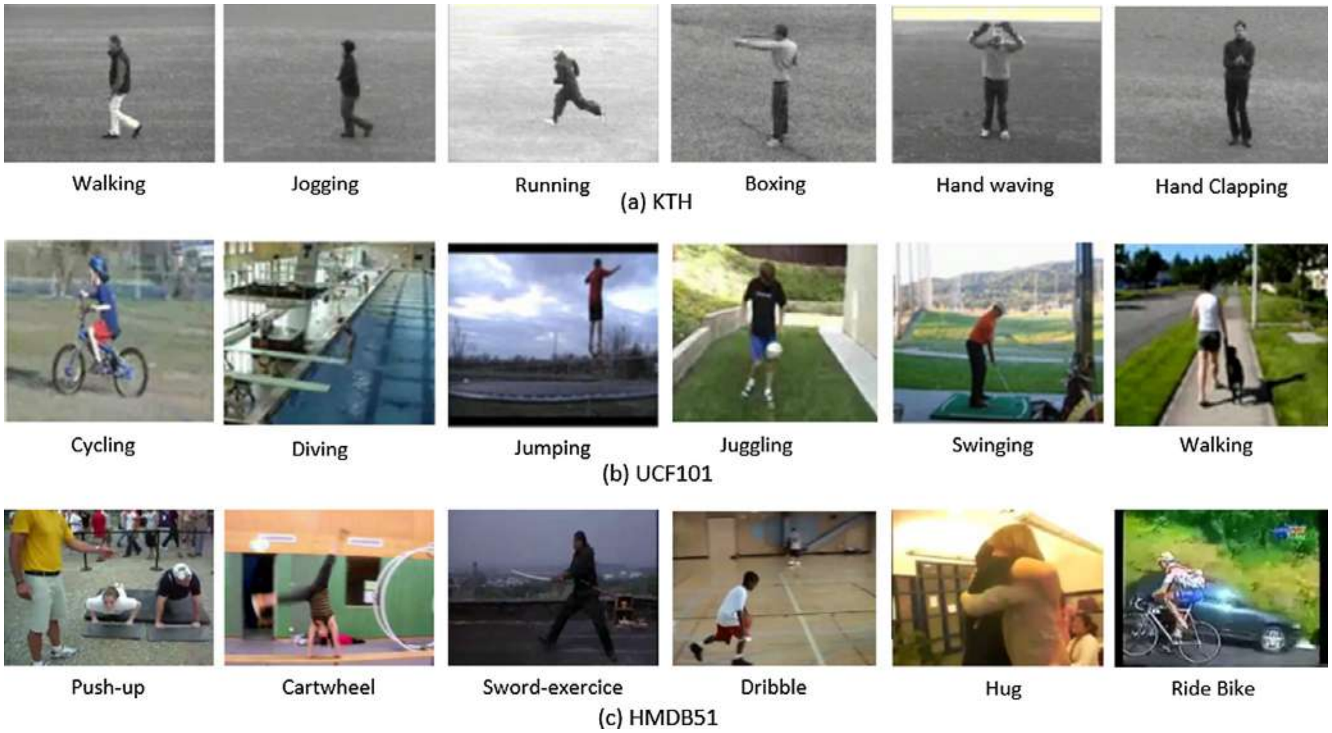

Fig. 11. Example frames from (a) KTH; (b) UCF101; (c) HMDB51.

HMDB51 [74]. HMDB51 is currently the largest dataset that addresses the problem of action recognition. It contains around 7000 manually annotated clips extracted from a variety of sources ranging from digitized movies to YouTube divided to 51 action categories. We follow the original protocol using three train test splits [74]. For every class and split, there are 70 videos for training and 30 videos for testing. Note that in our experiments, we use the original videos and not the stabilized ones.

6.2. Experimental settings

In the previous sections, we introduced the overall approach for motion segmentation and action description and recognition. Our proposed technique for motion segmentation does not require any assumptions about the first frame, initialization, or training steps. The segmentation process starts by dense SURF features extraction on a 6×6 sized grid with a temporal step size of N frames. In our experiments, we fix N to 3 so that small motions do not get lost and fast motions are captured without error. Then, L&K optical flow is computed. The flow vectors are clustered to determine whether camera motion exists. If it does not, a second clustering of flow vectors is conducted basing on the degree of similarity of their magnitudes, angles and closeness. Thresholds are fixed experimentally as follows: $l_{th} = 15, \theta_{th} = 2.0, posX_{th} = 45$ and $posY_{th} = 35$.

The descriptors employed in the action recognition process provide a rich video representation in terms of space and motion of moving interest points. From each clip, local spatio-temporal features as ST-SURF are extracted. As described previously, the extracted ST-SURF is a 68D vector (64D SURF, $\alpha_x, D_x, \alpha_y, D_y$). Square-shaped patches surrounding the detected SURFs are also extracted. The size of each detected patch is 20s. For each patch, a HMTO is computed in both planes (x, t) and (y, t). $HMTO_x$ and $HMTO_y$ are both 96D vectors. To reinforce our action recognition system, the motion boundary histogram MBH is used as a motion descriptor and as a remover of camera motion. MBH_x and MBH_y are 96D histograms. For both KTH and HMBD51 datasets, we follow the same protocols used in the methods of the state-of-the-art for learning and testing phases. A group of actors is involved during the learning phase. One actor, for every action, is left for the test step.

We performed an experiment using the bag-of-words approach to provide baseline results on the UCF101 dataset. The classification step starts by k-mean clustering applied to a set of 10^6 randomly selected features to build a visual dictionary for every extracted descriptor type (ST-SURF, $HMTO_x, HMTO_y, MBH_x, MBH_y$). For each one, we construct 4000 visual words. The k-mean clustering is initialized eight times, and we keep the configuration with the lowest error rate. The extracted histograms are L_2 normalized to ensure better visual quality. Finally, to classify the actions, we use a non linear SVM with an RBF_χ^2 Kernel [34].

$$K(v_i, v_j) = \exp\left(-\sum \frac{1}{A^c} D\left(v_i^c, v_j^c\right)\right), \tag{22}$$

where $D\left(v_i^c, v_j^c\right)$ is the χ^2 distance between video v_i and v_j of the channel c. A^c is the mean distance value of the training features.

6.3. Results and discussion

In this section, we report and discuss the motion segmentation and action recognition results reported from the three datasets. The purpose of this discussion is to highlight the key successes as well as the weaknesses of the proposed action recognition system.

6.3.1. Motion segmentation

We first present an evaluation of the proposed motion segmentation process. The experiments are carried out on 125 randomly picked videos (25 videos from each of the five categories) from the UCF dataset. This dataset is very complex. It represents different indoor and outdoor scenes with moving foregrounds, moving

Table 1
Processing time of motion segmentation over the UCF101 dataset.

Action class	Average duration (s)	Average processing time (s)
Sports	283	540.861
Playing Musical Instrument	89	127.516
Human-Object Interaction	156	289.502
Body-Motion Only	103	169.134
Human-Human Interaction	28	47.573

objects, complex backgrounds and camera motion. In fact, this dataset is dedicated mainly to the task of action recognition and, as far as we know, there are no evaluations of proposed motion segmentation algorithms based on this dataset to which we may compare our method.

The processing time of the overall algorithm for motion detection and segmentation for some videos from the five classes is presented in Table 1. The most time consuming task is computing dense features and optical flow on a small grid (6×6) every N frames. In fact, choosing a larger grid may accelerate the process but it decreases the system's performance.

The system's performance is evaluated in terms of the average F-measure given by:

$$F = \frac{2 \times R_c \times P_r}{R_c + P_r} \tag{23}$$

where P_r is precision and R_c is the recall for bounding box annotations, for each video. These measures are assessed based on certain bounding box annotations provided in [75]. Our main purpose in segmenting motion is to restrict the amount of data involved in studying human actions. Hence, we aim to detect a bounding box covering as much motion as possible. Table 2 reports the results obtained for the Sports (74.50%), Playing Musical Instrument (88.75%), Human-Object Interaction (87.83%), Body-Motion Only (85.45%), and Human-Human Interaction (84.53%) categories.

Generally, the camera motion segmentation process helps to improve the accuracy of motion segmentation. Furthermore, for fixed scenes, employing a second clustering of motion flow vectors enhances the extraction of moving objects. We consider these results satisfying, especially since we make no assumptions about the first frame, and our process does not require any initialization or training steps. Sports actions are considered to be the most challenging ones, as they include important motions of humans along with camera motion. The majority of these videos were captured outdoors with the presence of trees and audiences. Despite these effects, the motion segmentation of sports action achieved acceptable results. The performed motion segmentation makes the action recognition task easier, even with presence the of camera motion.

6.3.2. Action recognition

As previously mentioned, we used the same settings and evaluation metrics of the state-of-the-art in order to provide fair comparison. The accuracy rates reported on the KTH dataset are presented in Table 3. The distribution of the trajectory angles given by HMTO perform well in KTH dataset with a rate of 90.1% outperforming dense trajectory 89.8%, KLT trajectory 89.4% and SIFT trajectory 44.6%. As HMTO allows the tracking of the trajectory of a moving patch, the temporal extents of the action are settled by selective segmentation into actionlets. We also notice that AMAR-CTW [76] achieved 93.8% when they cluster motion curves using GMM in both learning/test steps. It could be also noticed that combined with ST-SURF and MBH, the HMTO gives best results in realistic and complicated video. This encourages the use of different features to achieve relevant action recognition. The results are improved by 4.9% on KTH dataset. This is the consequence of the efficiency of the association of the actionlets extraction with MBH features to reduce video in realistic benchmarks.

Table 2
F-measure results of motion segmentation over the UCF101 dataset.

Action class	F-measure (%)
Sports	74.50
Playing Musical Instrument	88.75
Human-Object Interaction	87.83
Body-Motion Only	85.45
Human-Human Interaction	84.53

Over the UCF101 dataset, the accuracy rates reported for the predefined action types are shown in Table 4: the Sports (87.23%), Playing Musical Instrument (79.4%), Human-Object Interaction (86.07%), Body-Motion Only (85.19%), and Human-Human Interaction (88.61%).

We observe that Human-Human Interaction actions achieve the highest accuracy, since the spatio-temporal segmentation we introduced in this work highlights human bodies. As a result, the feature extraction performed on the humans bounding boxes significantly boosts human detection. Analysis of the sports actions demonstrates a reasonable accuracy of 87.23%, which is due to two factors. The first factor is the temporal segmentation, while the second one is the motion based extraction features. Sports actions involve important motion, which can be very well described with our proposed approaches. Despite the fact that Human-Object and Body- Motion actions are not based on significant motion, the classification shows satisfactory results. We believe that the pixel motion segmentation precision in detecting motion is a good cue for exploring human action.

We present the results of our approach compared with the trajectory and motion-based video description approaches in Table 5. The MBH descriptor is associated with several approaches to detect human actions, since it is based on optical flow. This proves that combining MBH with different descriptors is a straightforward way to improve the results. The proposed approach, which combines ST-SURF, HMTO and MBH gives an accuracy rate of 79.2%, equivalent to the state-of-the-art trajectory-based video description. As expected, the proposed spatio-temporal segmentation improves the proposed approach by 6.9%, achieving 86.1% of accuracy in the challenging realistic large dataset UCF101. Compared with trajectory-based descriptors, the proposed approach performs well.

6.3.3. Comparison with the state of the art

The results given by the-state-of-the-art of KTH, UC101 and HMBD51 are given in Tables 6–8 respectively. Note that these results are reported from the papers in which they originally appeared.

The performances on the KTH dataset are around 94.9%, nearly 0.4% better than Spatio-temporal SURF [30]. We achieved 6–7% improvement more than the ST-SURF [6]. This is due to the optimization of the selective segmentation based on dense SURF and the fusion of the ST-SURF with trajectory descriptors (HMTO, MBH). The framework of the method proposed by Wang et al. [7] is similar to our framework but with performances that are a bit lower than ours. This dense trajectory approach [7] starts by sampling dense points from each frame, then tracks them and estimates their displacements as trajectory information from an optical flow field. After that, this trajectory information is encoded to classify different actions. However, the number of sampling points is significantly greater than in our method. Even though, this approach achieves good recognition results, it is

Table 3
Results of various descriptors performances in action recognition over the KTH dataset.

	Proposed approach				KLT		SIFT		CTW
Descriptor	HMTO	ST-SURF	MBH	Fused	Traj	Fused	Traj	Fused	Traj
Accuracy (%)	90.1	88.2	90	**94.9**	89.4	93.4	44.9	84.9	93.8

Table 4
Recognition results of the proposed approach over the UCF101 dataset.

Action class	Accuracy (%)
Sports	87.23
Playing Musical Instrument	83.4
Human-Object Interaction	86.07
Body-Motion Only	85.19
Human-Human Interaction	88.61
Average	86.1

Table 5
Trajectory based descriptor performances over the UCF101 dataset (Traj: Trajectory; LocDesc: Local descriptors).

Approach	Descriptor	Accuracy (%)
Traj	TrajShape	47.1
Traj + LocDesc	TrajShape + MBH + HOG + HOF	72.8
LocDesc	HOG3D + MBH + HOG + HOF	78.9
Traj	Dense trajectory	85.9
Traj	Dense trajectory + PSIFT	85.7
Traj	MBH	85.7
Traj	ST-SURF + HMTO + MBH	79.2
Traj	proposed: BB + ST-SURF + HMTO + MBH	**86.1**

Table 6
Comparison with the state-of-the-art approaches over the KTH dataset.

Method	Accuracy (%)
Shuldt et al. [73]	71.7
Jhuang et al. [41]	90.5
Laptev et al. [34]	91.8
Niebles et al. [39]	93.3
Lin et al. [78]	95.8
Noguchi et al. [30]	94.5
Megrhi et al. [6]	88.2
Wang et al. [7]	94.2
Virigkas et al. [76] (CTW)	93.8
Ni et al. [77]	95
Proposed	**94.9**

Table 7
Comparison with the state-of-the-art approaches over the UCF101 dataset.

Method	Accuracy (%)
Murthy and Goecke [79]	72.8
Shi et al. [80]	78.9
Wang and Schmid [81]	85.9
Karaman et al. [82]	85.7
Soomro et al. [71]	44.5
Karpathy et al. [83]	63.3
Simonyan et al. [84]	87.6
Proposed	**86.1**

Table 8
Comparison with the state-of-the-art approaches over the HMDB51 dataset.

Method	Accuracy (%)
Wang et al. [85]	55.9
Jiang et al. [87]	40.7
Murthy and Goecke [79]	47.3
Shi et al. [80]	55.2
Ni et al. [77]	52.3
Jain et al. [22]	52.1
Ballas et al. [86]	51.8
Proposed	**58.8**

computationally very complex. Far from interest points-based methods, Ni et al. [77] use human pose information as the main cue to select the most representative sub-volume in the video to recognize actions. This method achieved an accuracy of 95% which is almost equal to our results. This proves that motion and appearance descriptor are as important as human pose to describe the action.

From Table 7, the overall performances on UCF101 dataset are 86.1%. These results are significantly better than those reported in [71] using the standard bag-of-words method with an overall accuracy of 44.5%. In [79], the authors used a dense trajectory computed for fixed frame length $L = 16$ and $L = 17$. The overall performance rate is 47.1% using a trajectory descriptor. Combined with MBH, HOG and HOF their trajectory-based approach reaches 72.8%. These performances are still less satisfactory than our results. We believe that this is due to their use of a fixed frame number. We also outperform the results given in [80]. The authors of this paper used a multi-channel approach for the Local Part Model and the LPM algorithm for efficient action recognition. Their approach was based on the fusion of HOG, HOF, HOG3D and MBH. They achieved 78.9% of average accuracy. Dense trajectory features were used in [81]. The author applied Fisher vectors and spatio-temporal pyramids to embed structural information. Finally, a linear SVM combining all their descriptors gives a performance of 85.9%, about 0.2% lower than our results. This proves the importance of motion cues in detecting human actions. This also encourages us to investigate more approaches than the SVM we used in this work. Fisher vectors give good results in [82]. In that study, the authors extracted features from both video and key frame modalities. They used dense trajectory features associated with HOG and Motion Boundary Histogram (MBH). Next, they encoded them as Fisher vectors. To represent action-specific scene context, we compute local SIFT pyramids on grayscale (P-SIFT) and opponent color keyframes (P-OSIFT) extracted as the central frame of each clip. They proposed to improve accuracy by using L1-regularized logistic regression (L1LRS) for stacking classifier outputs 85.7%, 0.4% lower than our method. The results given in [83] are less satisfactory than ours. These authors provide an extensive empirical evaluation of CNNs on large-scale video classification 63.3%. However, in [84], authors investigate architectures of indiscriminately trained deep Convolutional Networks (ConvNets) for action recognition in video. This method achieves 87.6% accuracy, which is the best result of all the approaches studied. The success of this method also highlights the importance of the classification task investigation, especially in terms of deep classification.

Finally, comparison results with the state-of-the-art approaches over the HMDB51 dataset are presented in Table 8. Until now, this dataset is considered as the most challenging one. In fact, efforts are being made to achieve high performances on it. With our proposed technique, we succeeded to achieve an accuracy rate of 58.82% which is an important score compared to the state of the art as presented in Table 8. Wang et al. [85] achieve 55.9% over the baseline videos (without motion stabilization) basing on improved dense trajectories with Fisher Vector encoding. This method include local appearance (HOG) and motion descriptors (HOF/MBH). Ballas et al. [86] achieved 51:8% by pooling dense trajectory features from regions of interest using video structural cues estimated by different saliency functions. Jiang et al. [87] achieve 40.7% by modeling the relationship between dense trajectory clusters. Our method surpasses again the methods proposed by Murthy and Goecke [79], Shi et al. [80] and Ni et al. [77].

It can also be noticed that the performances on the KTH are better than those on UCF101 and HMDB51. This is due to the fact that KTH is a controlled dataset with minimum camera motion.

7. Conclusion

In this paper, we presented an end-to-end framework for human action recognition in big datasets. As part of this effort, we started by introducing a new human action segmentation process. Our method is based on studying optical flows induced by human motion which are, then, clustered to determine the existence of camera motion. The latter, if it exists, is compensated by means of affine transformation. Finally, human motion is extracted using temporal differencing along with pre-processing operations to reduce noise. Our second contribution in this framework is the video description process. It is a combination of motion, trajectory and appearance descriptors. To this end, the actions classification task is achieved using a support-vector-machine to classify actions based on extracted features by means of a bag-of-words approach. We have shown promising results in both action detection and recognition processes in videos taken under different conditions and with complex background. Compared to many existing state-of-the-art approaches, our proposed framework achieves a reasonable trade-off between high accuracy and prohibitive computational cost.

Acknowledgements

The research and innovation work of the second author is carried out within a MOBIDOC thesis funded by the European Union under the PASRI project and administered by the ANPR.

References

[1] T. Brox, J. Malik, Object segmentation by long term analysis of point trajectories, in: Computer Vision–ECCV 2010, Springer, 2010, pp. 282–295.
[2] A. Gaidon, Z. Harchaoui, C. Schmid, Recognizing activities with cluster-trees of tracklets, in: BMVC 2012-British Machine Vision Conference, BMVA Press, 2012, pp. 30–31.
[3] N.P. Cuntoor, R. Chellappa, Epitomic representation of human activities, in: Computer Vision and Pattern Recognition, CVPR'07, IEEE, 2007, pp. 1–8.
[4] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, J. Li, Hierarchical spatio-temporal context modeling for action recognition, in: Computer Vision and Pattern Recognition, CVPR'09, IEEE, 2009, pp. 2004–2011.
[5] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Dense trajectories and motion boundary descriptors for action recognition, Int. J. Comput. Vision (2013) 1–20.
[6] S. Megrhi, W. Souidène, A. Beghdadi, Spatio-temporal surf for human action recognition, in: Advances in Multimedia Information Processing–PCM 2013, Springer, 2013, pp. 505–516.
[7] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Action recognition by dense trajectories, in: Computer Vision and Pattern Recognition (CVPR'11), IEEE, 2011, pp. 3169–3176.
[8] L. Shao, L. Ji, Y. Liu, J. Zhang, Human action segmentation and recognition via motion and shape analysis, Pattern Recogn. Lett. 33 (4) (2012) 438–445.
[9] L. Zappella, X. Lladó, J. Salvi, Motion segmentation: a review, in: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence, IOS Press, 2008, pp. 398–407.
[10] L. Zappella, X. Lladó, J. Salvi, New trends in motion segmentation, Pattern Recogn. (2009) 31–46.
[11] T. Brox, M. Rousson, R. Deriche, J. Weickert, Colour, texture, and motion in level set based segmentation and tracking, Image Vis. Comput. (2010) 376–390.
[12] B.K. Horn, B.G. Schunck, Determining optical flow, in: Technical Symposium East, International Society for Optics and Photonics, 1981, pp. 319–331.
[13] B.D. Lucas, T. Kanade, et al., An iterative image registration technique with an application to stereo vision, IJCAI, vol. 81, 1981, pp. 674–679.
[14] J.-Y. Bouguet, Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm, Intel Corporation 5 (1–10) (2001) 4.
[15] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[16] H. Bay, T. Tuytelaars, L. Van Gool, Surf: speeded up robust features, in: Computer vision–ECCV 2006, Springer, 2006, pp. 404–417.
[17] F. Jurie, B. Triggs, Creating efficient codebooks for visual recognition, in: Tenth IEEE International Conference on Computer Vision, ICCV'05, vol. 1, IEEE, 2005, pp. 604–610.
[18] J.R. Uijlings, A.W. Smeulders, R.J. Scha, Real-time visual concept classification, IEEE Trans. Multimedia 12 (7) (2010) 665–681.
[19] L. Fei-Fei, P. Perona, A bayesian hierarchical model for learning natural scene categories, Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 2, IEEE, 2005, pp. 524–531.
[20] H. Wang, M.M. Ullah, A. Klaser, I. Laptev, C. Schmid, et al., Evaluation of local spatio-temporal features for action recognition, in: BMVC'09 – British Machine Vision Conference, 2009.
[21] J.-M. Odobez, P. Bouthemy, Robust multiresolution estimation of parametric motion models, J. Visual Commun. Image Representation 6 (4) (1995) 348–365.
[22] M. Jain, H. Jégou, P. Bouthemy, Better exploiting motion for better action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13), IEEE, 2013, pp. 2555–2562.
[23] S. Wu, O. Oreifej, 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 (ICCV'11), IEEE, 2011, pp. 1419–1426.
[24] W. Li, Z. Zhang, Z. Liu, Expandable data-driven graphical modeling of human actions based on salient postures, IEEE Trans. Circuits Syst. Video Technol. (2008) 1499–1510.
[25] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'05, vol. 1, IEEE, 2005, pp. 886–893.
[26] G. Willems, T. Tuytelaars, L. Van Gool, An efficient dense and scale-invariant spatio-temporal interest point detector, in: Computer Vision–ECCV'08, Springer, 2008, pp. 650–663.
[27] I. Laptev, On space-time interest points, Int. J. Comput. Vis. 64 (2–3) (2005) 107–123.
[28] A. Klaser, M. Marszałek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, in: BMVC 2008-19th British Machine Vision Conference, British Machine Vision Association, 2008, pp. 275-1.
[29] P. Scovanner, S. Ali, M. Shah, A 3-dimensional sift descriptor and its application to action recognition, in: Proceedings of the 15th International Conference on Multimedia, ACM, 2007, pp. 357–360.
[30] A. Noguchi, K. Yanai, A SURF-based spatio-temporal feature for feature-fusion-based action recognition, in: Trends and Topics in Computer Vision, Springer, 2012, pp. 153–167.
[31] J. Sun, Y. Mu, S. Yan, L.-F. Cheong, Activity recognition using dense long-duration trajectories, in: IEEE International Conference on Multimedia and Expo (ICME'10), IEEE, 2010, pp. 322–327.
[32] S. Megrhi, Spatio-temporal Descriptors for Human Action Detection and Recognition, Ph.D. Thesis, University Paris 13, 2015.
[33] A. Gaidon, Z. Harchaoui, C. Schmid, Actom sequence models for efficient action detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), IEEE, 2011, pp. 3201–3208.
[34] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR'08, IEEE, 2008, pp. 1–8.
[35] S. Megrhi, M. Jmal, A. Beghdadi, W. Mseddi, Spatio-temporal action localization for human action recognition in large dataset, in: IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics, 2015, 94070O–94070O.
[36] N. Ikizler-Cinbis, S. Sclaroff, Object, scene and actions: combining multiple features for human action recognition,;;, in: Computer Vision–ECCV'10, 2010, pp. 494–507.
[37] D.H. Nga, K. Yanai, A spatio-temporal feature based on triangulation of dense SURF, in: IEEE International Conference on Computer Vision Workshops, ICCVW'13, IEEE, 2013, pp. 420–427.
[38] G. Piriou, P. Bouthemy, J.-F. Yao, Recognition of dynamic video contents with global probabilistic models of visual motion, IEEE Trans. Image Process. 15 (11) (2006) 3417–3430.
[39] J.C. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human action categories using spatial-temporal words, Int. J. Comput. Vis. 79 (3) (2008) 299–318.
[40] P. Dollár, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, IEEE, 2005, pp. 65–72.
[41] H. Jhuang, T. Serre, L. Wolf, T. Poggio, A biologically inspired system for action recognition, in: IEEE 11th International Conference on Computer Vision, ICCV'07, IEEE, 2007, pp. 1–8.
[42] L. Yeffet, L. Wolf, Local trinary patterns for human action recognition, in: IEEE 12th International Conference on Computer Vision, ICCV'2009, IEEE, 2009, pp. 492–497.

[43] P. Matikainen, M. Hebert, R. Sukthankar, Trajectons: action recognition through the motion analysis of tracked features, in: IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), IEEE, 2009, pp. 514–521.

[44] R. Messing, C. Pal, H. Kautz, Activity recognition using the velocity histories of tracked keypoints, in: IEEE 12th International Conference on Computer Vision, ICCV'09, IEEE, 2009, pp. 104–111.

[45] C. Fanti, L. Zelnik-Manor, P. Perona, Hybrid models for human motion recognition, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'05, vol. 1, IEEE, 2005, pp. 1166–1173.

[46] O. Oreifej, Z. Liu, Hon4d: histogram of oriented 4d normals for activity recognition from depth sequences, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR'13, IEEE, 2013, pp. 716–723.

[47] N. Ikizler, R.G. Cinbis, P. Duygulu, Human action recognition with line and flow histograms, in: 19th International Conference on Pattern Recognition, ICPR'08, IEEE, 2008, pp. 1–4.

[48] Y. Song, L. Goncalves, P. Perona, Unsupervised Learning of Human Motion Models, MIT Press, 2002.

[49] C. Rao, A. Yilmaz, M. Shah, View-invariant representation and recognition of actions, Int. J. Comput. Vis. (2002) 203–226.

[50] H. Uemura, S. Ishikawa, K. Mikolajczyk, Feature tracking and motion compensation for action recognition, in: BMVC, 2008, pp. 1–10.

[51] N. Johnson, D. Hogg, Learning the distribution of object trajectories for event recognition, Image Vis. Comput. (1996) 609–615.

[52] W.-C. Lu, Y.-C. Wang, C.-S. Chen, Learning dense optical-flow trajectory patterns for video object extraction, in: Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS'10, IEEE, 2010, pp. 315–322.

[53] A. Kläser, M. Marszałek, C. Schmid, A. Zisserman, Human focused action localization in video, in: Trends and Topics in Computer Vision, Springer, 2012, pp. 219–233.

[54] I. Laptev, P. Pérez, Retrieving actions in movies, in: IEEE 11th International Conference on Computer Vision, ICCV'07, IEEE, 2007, pp. 1–8.

[55] D.J. Fleet, A.D. Jepson, Stability of phase information, IEEE Trans. Pattern Anal. Mach. Intell. (1993) 1253–1268.

[56] P. Viola, M.J. Jones, D. Snow, Detecting pedestrians using patterns of motion and appearance, in: Ninth IEEE International Conference on Computer Vision, ICCV'03, IEEE, 2003, pp. 734–741.

[57] O. Kliper-Gross, Y. Gurovich, T. Hassner, L. Wolf, Motion interchange patterns for action recognition in unconstrained videos, in: Computer Vision–ECCV'12, Springer, 2012, pp. 256–269.

[58] N. Dalal, B. Triggs, C. Schmid, Human detection using oriented histograms of flow and appearance, in: Computer Vision–ECCV'06, Springer, 2006, pp. 428–441.

[59] F. Moosmann, B. Triggs, F. Jurie, et al., Fast discriminative visual codebooks using randomized clustering forests, NIPS, vol. 2, 2006, p. 4.

[60] T.-H. Yu, T.-K. Kim, R. Cipolla, Real-time action recognition by spatiotemporal semantic and structural forests, in: Proceedings of the British Machine Vision Conference, 2010, p. 56.

[61] T. Guha, R.K. Ward, Learning sparse representations for human action recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2012) 1576–1588.

[62] K.N. Tran, I.A. Kakadiaris, S.K. Shah, Modeling motion of body parts for action recognition, in: BMVC'11, Citeseer, 2011, pp. 1–12.

[63] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Workshop on Statistical Learning in Computer Vision, ECCV'04, 2004, p. 22.

[64] A.P. Brandão Lopes, E. Alves do Valle Jr, J. Marques de Almeida, A. Albuquerque de Araújo, Action Recognition in Videos: from Motion Capture Labs to the Web. Available from: <1006.3506>.

[65] J. Liu, J. Luo, M. Shah, Recognizing realistic actions from videos in the wild, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR'09, IEEE, 2009, pp. 1996–2003.

[66] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'06, vol. 2, IEEE, 2006, pp. 2169–2178.

[67] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. (1986) 679–698.

[68] J.D. Foley, A. Van Dam, S.K. Feiner, J.F. Hughes, R.L. Phillips, Introduction to Computer Graphics, vol. 55, Addison-Wesley Reading, 1994.

[69] R. Brunelli, Template Matching Techniques in Computer Vision, 2008.

[70] D. Sun, S. Roth, M.J. Black, Secrets of optical flow estimation and their principles, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR'10, IEEE, 2010, pp. 2432–2439.

[71] K. Soomro, A.R. Zamir, M. Shah, UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild, CoRR abs/1212.0402. URL <http://arxiv.org/abs/1212.0402>.

[72] Youtube, Statistiques, 2009. <https://www.youtube.com/yt/press/fr/statistics.html/>.

[73] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local SVM approach, Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 3, IEEE, 2004, pp. 32–36.

[74] H. Kuehne, H. Jhuang, R. Stiefelhagen, T. Serre, HMDB51: a large video database for human motion recognition, in: High Performance Computing in Science and Engineering 12, Springer, 2013, pp. 571–582.

[75] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, R. Sukthankar, THUMOS Challenge: Action Recognition with a Large Number of Classes, 2014. <http://crcv.ucf.edu/THUMOS14/>.

[76] M. Vrigkas, V. Karavasilis, C. Nikou, I.A. Kakadiaris, Matching mixtures of curves for human action recognition, Comput. Vis. Image Underst. 119 (2014) 27–40.

[77] B. Ni, P. Moulin, S. Yan, Pose adaptive motion feature pooling for human action analysis, Int. J. Comput. Vis. 111 (2) (2015) 229–248.

[78] Z. Lin, Z. Jiang, L.S. Davis, Recognizing actions by shape-motion prototype trees, in: IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 444–451.

[79] O. Murthy, R. Goecke, Ordered Trajectories for Large Scale Human Action Recognition, in: International Conference on Computer Vision Workshops (ICCVW), IEEE, 2013, pp. 412–419.

[80] F. Shi, R. Laganiere, E. Petriu, H. Zhen, Lpm for fast action recognition with large number of classes, in: THUMOS: ICCV Workshop on Action Recognition with a Large Number of Classes. Notebook paper, 2013.

[81] H. Wang, C. Schmid, LEAR-INRIA submission for the THUMOS workshop, in: ICCV Workshop on Action Recognition with a Large Number of Classes, 2013.

[82] S. Karaman, L. Seidenari, A.D. Bagdanov, A. Del Bimbo, L1-regularized logistic regression stacking and transductive crf smoothing for action recognition in video, ICCV Workshop on Action Recognition with a Large Number of Classes, vol. 13, 2013, p. 14.

[83] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2014, pp. 1725–1732.

[84] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in Neural Information Processing Systems, 2014, pp. 568–576.

[85] H. Wang, D. Oneata, J. Verbeek, C. Schmid, A robust and efficient video representation for action recognition, Int. J. Comput. Vis. (2015) 1–20.

[86] N. Ballas, Y. Yang, Z.-Z. Lan, B. Delezoide, F. Prêteux, A. Hauptmann, Space-time robust representation for action recognition, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2704–2711.

[87] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, C.-W. Ngo, Trajectory-based modeling of human actions with motion reference points, in: European Conference on Computer Vision, Springer, 2012, pp. 425–438.