Spatio-Temporal Action Localization For Human Action Recognition in Large Dataset

Sameh MEGRHI[1], Marwa JMAL [2], Azeddine BEGHDADI[1] and Wided Mseddi[1,2]

[1]L2TI, Institut Galilée, Université Paris 13, France;
[2]SERCOM, Ecole Polytechnique de Tunisie

ABSTRACT

Human action recognition has drawn much attention in the field of video analysis. In this paper, we develop a human action detection and recognition process based on the tracking of Interest Points (IP) trajectory. A pre-processing step that performs spatio-temporal action detection is proposed. This step uses optical flow along with dense speed-up-robust-features (SURF) in order to detect and track moving humans in moving field of views. The video description step is based on a fusion process that combines displacement and spatio temporal descriptors. Experiments are carried out on the big data-set UCF-101. Experimental results reveal that the proposed techniques achieve better performances compared to many existing state-of-the-art action recognition approaches.

Keywords: Action recognition, trajectory, dense sampling, Optical flow, spatio-temporal descriptor, displacement descriptor, fusion.

1. INTRODUCTION

Human action detection and recognition in real scenes remains an important topic in computer vision. However, it is a challenging task as videos may include complex actions with large intra-class variations, poor quality or camera motion. An efficient remedy to these challenges is a relevant video description thus segmentation of videos into small sequences. In order to segment video sequences and reduce the amount of data involved in this task, we propose in this work to detect and segment human motion. Human motion segmentation is another theme in recent work.[1] Pixel-wise techniques, namely background subtraction and temporal differencing,[2] are the most straightforward methods in this task. However, when dealing with moving cameras, these models are likely to fail as the background is continuously varying along with the target's motion. Optical flow (OF) based methods[3] are one of the most employed techniques in motion segmentation. Lucas and Kanade (L&K)[4] is one of the oldest yet most employed OF algorithms. Regarding its limitations toward accuracy and illumination changes, some improvements are presented.[5] In our research, to detect moving objects, we computed the OF of IP extracted using SURF[6] descriptors over a regular dense grid. In fact, a recent evaluation of dense sampling proposed by Uijlings et al.[7] proved that dense SURF descriptors may be extracted more quickly with no loss of accuracy.

Among the major obstacles facing robust motion segmentation is camera motion. In this case, objects' motion is combined with both the camera and background motions. Thus, camera motion compensation is compulsory. Earlier approaches relied on estimating the camera motion as a 2D affine transform or homography.[8,9] Cinbis *et al.*[10] applied video stabilization using homography-based motion compensation approach. Nga *et al.*[11] subtracted the estimated camera flow, multiplied by the camera direction, from the flow of each extracted spatio-temporal keypoint. 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.

To describe actions in videos, spatio-temporal (ST) local features were widely exploited.[12] In the work of Dalal et al.,[13] ST descriptors are extracted by extending the 2D IP to the temporal domain (1D). Laptev

Further author information:
S.M.: E-mail: sameh.megrhi@univ-paris13.fr
M.J.: E-mail: marwa.jmal@telnet-consulting.com
A.B.: E-mail: azeddine.beghdadi@univ-paris13.fr
W.M.: E-mail: wided.mseddi@univ-paris13.fr

and al.,[14] expanded the volumetric features corner detector to extract space-time local structures. Local descriptors were also extended to the temporal domain such as the histograms of oriented 3D spatio-temporal gradients,[15] E-SURF[16] and the 3D-SIFT.[17] In,[18] it has been 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 in a new scheme deserves to be more investigated and might be the cue of success for action detection in big datasets. That is to detect spatio-temporal features; various works tracked IP upon a video sequence. Indeed, Sun et al.,[19] performed efficient action recognition by leveraging the motion information of trajectories. Sameh et al.,[20] proposed a method based on tracking the trajectory of SURF IPs into a frame packet. One of the latest work is proposed in[18] where descriptors based on appearance (Histogram of oriented gradient), motion (Histogram of optical flow) and trajectories are fused to characterize shape (point coordinates). Their approach provided excellent performances for action recognition. What we propose is also based on trajectory tracking. We suggest a video description based on the Spatio-temporal SURF (ST-SURF),[20] the histogram of motion trajectory orientation (HMTO) and the motion boundary histogram (MBH).[13]

2. RELATED WORKS

Recently, researchers are focusing on action description by tracking interest points motion.[21] This allows exploring several motion cues such as velocity,[22] orientation,[23] location,[24] trajectory curves,[25] trajectory parts[26] or different motion cues combinations.[20] Moreover, Sun *et al.*,[27] 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[4] which is commonly employed in videos.[28] Authors in[19] used both SIFT and KLT features to extract long duration trajectories. Trajectory tracking is proven to be an intuitive and successful approach in several public datasets.[18] Trajectory segmentation is another difficult task for trajectories description. To segment the trajectory, several works are based on trajectory clustering,[29] others on moving object trajectory tracking. More recently, in,[18] 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 frame. Finally, a trajectory shape descriptor which characterizes the displacement is computed.

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.[30] Although, many schemes have been proposed to reduce camera motion,[31] 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.[32] It removes constant motion and preserves significant one. MBH was employed in various action recognition schemes.[18] It provides more interesting results when applied to video containing important camera motion. However, MBH is not dedicated to remove camera motion, but combined with the spatio-temporal SURF (ST-SURF) proposed by,[20] 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,[33] Sparse code-book learning[34,35] or bag of visual words (BOVW).[18,36] The BOVW approach achieved good results in action recognition in both image[37] and video analysis.[38] 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.

3. HUMAN MOTION DETECTION AND SEGMENTATION

The proposed motion segmentation algorithm is based on computing optical flows of detected dense features. Moreover, camera motion estimation is achieved by tracking features between successive frames: first interest keypoints are extracted using dense SURF, then, the iterative Lucas & Kanade (LK) OF using a pyramidal

representation[5] is applied to track them over the next frame. However, in an image, several parts, such as the sky or the roof, have almost the same color distribution and do not, generally, contribute with useful information. They also may add noise in the estimated OF. In order to overcome this drawback while preserving the most important structural features, we start our process by detecting image edges using canny edge detector.[39]As follows, all steps of the motion segmentation process will be applied on the edge frame. OF computation outputs a set of four-dimensional vectors V such as:

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

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 in the current frame and its corresponding feature in the next one. In general, OF is computed between two successive frames. However, the result may be unstable when objects either move too fast, too slowly or stop between two frames. In this paper, we propose to extract keypoints and compute OF with a temporal step size of N frames. This step allows also the removal of static features that correspond to pixels with OF magnitudes lower than a threshold T in both x and y directions. In our experiments, the minimum motion magnitude is set to 0.5 pixel per frame.

3.1 Detection and compensation of camera motion

Camera motion detection : Once dense features are tracked, we should verify the existence of camera motion by analyzing OFs between two frames in a frame set under the assumption that if most points move in the same direction, camera motion exists and has the same direction as the moving points. At this level, we propose to cluster OF vectors 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 flow directions. In this work, we define eight possible directions for the camera: 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). The flow field is segmented into these 8 clusters by employing k-Nearest Neighbor (KNN) clustering algorithm. However, some small clusters that do not belong to a dominant cluster may appear. To reduce unuseful data, clusters with a size lower than a certain threshold are discarded. Figure 1 presents examples of OF clustering using KNN. In these representations, it is easy to distinguish the moving objects from the background as well as determining the direction of camera motion. Thereafter, the size of each of the eight clusters is determined

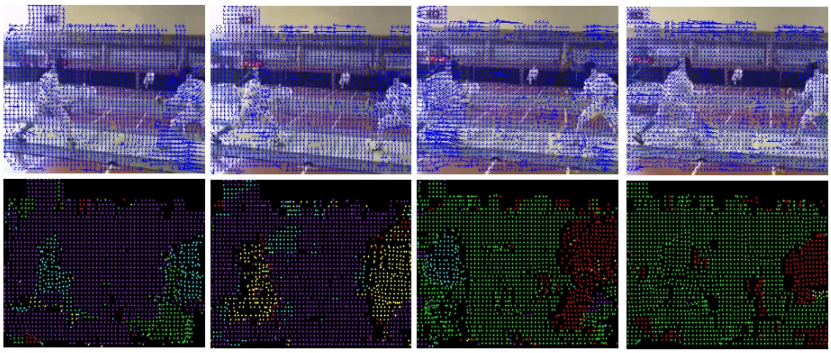

Figure 1. OF clustering using KNN algorithm. Row 1: OFs between two successives frames. Row 2: Results of KNN clustering. Each of the eight directions of the camera is represented by a different color.

and compared to a threshold. Finally, camera motion exits if $\sup_{cl \in \{1,..,8\}}\{s_{cl}\} \geqslant K$; Where s_{cl} is the size of cluster cl and K is a threshold representing the minimal required proportion of moving points. In our experiments, we set K as $\frac{N}{2}$ where N is the total number of detected points. The motion magnitude and deviation are then computed as follows:

$$m_m = mean \mid f_i \mid \quad ; \quad \theta_m = mean(\theta_{f_i}) \tag{2}$$

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

Camera motion Compensation : in videos captured by a hand-held camera, camera motion is random.

This motion is a combination of translation and rotation. In Nga *et al.*'s work,[11] 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.

In this work, we take under consideration only translation and rotation motions. Scaling is one of our future works.Hence, the transformed frame I' is obtained as follows:

$$I' = \begin{bmatrix} \cos\theta_m & -\sin\theta_m \\ \sin\theta_m & \cos\theta_m \end{bmatrix} \times I + \begin{bmatrix} m_{mH} \\ m_{mV} \end{bmatrix} \tag{3}$$

Here I is the original frame; m_{mH} (respectively m_{mV}) refers to the camera flow magnitude when the camera translates horizontally (respectively vertically). d equals 1 if the camera moves in a positive direction or -1 if the camera moves in a negative direction. In case of horizontal motion, $m_{mV} = 0$ and in case of vertical motion, $m_{mH} = 0$. Unlike in[9] and[11] where the motion of each flow vector is compensated for independently, in our work, we apply the affine model on the whole image.

3.2 Foreground extraction

After compensating the camera motion, we obtain a situation similar to one where the camera is static. Here, moving objects are segmented using a pixel-wise technique known as temporal differencing. It is the simplest method for extracting moving objects and is robust in dynamic environments. This algorithm classifies a new pixel as being a foreground pixel whenever $\parallel I(x,y) - I_{prev}(x,y) \parallel \geq T_h$ where T_h is a defined threshold. The obtained result is a binary image. However, due to camera noise and limitations of the background model, the foreground mask (binary image) typically contains numerous small "noise" clusters. These erroneous clusters can be removed by applying a noise filtering algorithm to the foreground mask. In fact, 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, both morphological erosion and dilatation, respectively, are employed to remove noise and unwanted objects. 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

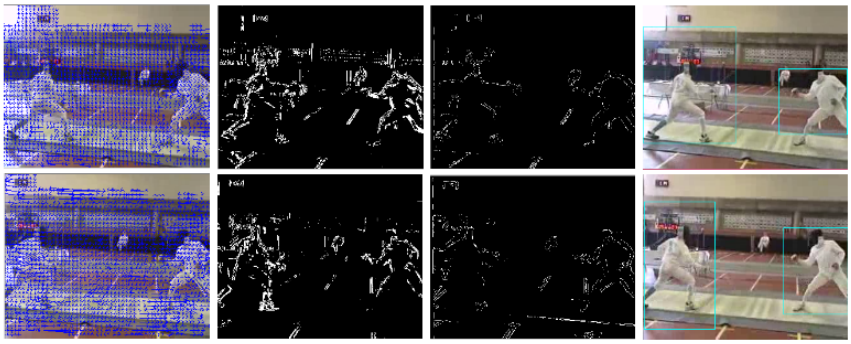

Figure 2. Results of our proposed method for motion segmentation. Column 1: consecutive frames from a video sequence with camera motion on which OF is drawn. Column 2: segmentation results before camera motion compensation. Column 3: segmentation results after camera motion compensation. Column 4: final segmentation results.

of our proposed method for motion segmentation are applied to an input video with a temporal step of size N. Thus, the detected objects need to be tracked in the remaining frames. To accomplish this, a template matching technique, normalized cross correlation,[40] is employed. Figure 2 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 foreground. Our method succeeded to eliminate the motion induced by the camera and thus leaving only the motion of humans/objects. If the camera is detected as being still, we admit that the detected flow belongs to the objects/humans in motion. Hence, instead of applying the temporal differencing technique, here, a second clustering of OF vectors is applied based on

the degree of similarity of their magnitudes, angles and closeness, under the assumption that OFs of a single person/object have similar characteristics. We assume that two OF vectors, f_i and f_j, belong to the same cluster if the following assumptions are satisfied:

$$| l_i - l_j | \leq l_{th} \quad ; \quad | \theta_i - \theta_j | \leq \theta_{th} \quad ; \quad | posX_i - posX_j | \leq posX_{th} \quad ; \quad | posY_i - posY_j | \leq posY_{th} \tag{4}$$

where l_{idx} is the magnitude of f_{idx}; θ_{idx} is the deviation (angle) of f_{idx}; (X_{idx}, Y_{idx}) are the coordinates of the OF vectors. Finally, l_{th}, θ_{th}, $posX_{th}$ and $posY_{th}$ are thresholds.
All detected flow vectors are compared two-by-two based on these similarity comparisons leading to form a fixed number of clusters. In order to remove noisy and meaningless clusters, we discard ones with a size smaller than a threshold. The remaining clusters belong to the foreground. A bounding box is drawn around each one. Figure 3 presents the segmentation results derived from the OF clustering technique compared to

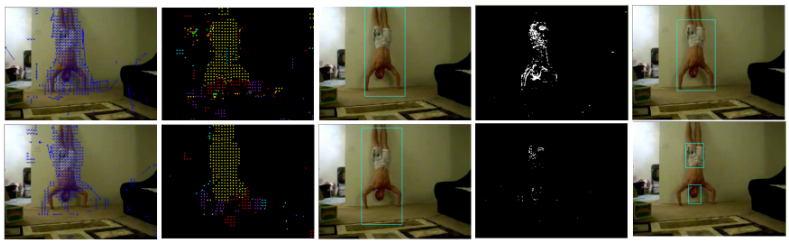

Figure 3. Motion segmentation in videos with a static camera. Column 1: a set of consecutive frames. Columns 2 and 3: OF clustering using KNN algorithm. Columns 4 and 5: results of the temporal differencing technique.

using the frame differencing technique. The OF clustering technique (column 3) achieves better segmentation results. It succeeds to capture the whole human motion, whereas the second technique (column 5) leads to loss of information and only some parts of the motion are segmented.

4. HUMAN ACTION RECOGNITION FRAMEWORK

In order to detect actions, video are not usually treated as whole part. In this paper, we propose to segment videos based on the actions or sub-actions they contain. For a relevant spatial-temporal video segmentation into small video portions, we extract IP located into the Bounding box. The trajectory of the extracted descriptor is then tracked until the end of the video segment to be analyzed. The steps for selective temporal segmentation are described below:

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

One of the main objectives of the proposed method is to reduce the number of video frames to be treated. We propose the use of concepts of selective snippets and the group of SURF (G-SURF). 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 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 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 $(n+2)$.

$$D_{n,n+1} = (Dx_{n,n+1}, Dy_{n,n+1}, Dt_{n,n+1}) \tag{5}$$

and

$$D_{n+1,n+2} = (Dx_{n+1,n+2}, Dy_{n+1,n+2}, Dt_{n+1,n+2}) \tag{6}$$

$$\alpha = \arccos \frac{D_{n,n+1} \cdot D_{n+1,n+2}}{\|D_{n,n+1}\| \times \|D_{n,n+2}\|} \tag{7}$$

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}. 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}.

4.2 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 extraction of the used descriptors. The motion trajectory detection tracking and extraction are based on the following steps:

4.2.1 Optical flow extraction

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[41] have been proposed. Sun et al.[41] 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.[41] 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.[41] The initially computed optical flow serves in many blocks in the proposed framework. This reduces feature extraction computational time.

4.2.2 Trajectory tracking

To every selective snippet corresponds 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.[41] 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 α_x between optical flow and the plane (t, x), the angle α_y between the plane (t, y) and the motion vector.

$$\propto_x = 90 - \frac{180}{\Pi}\arctan(\frac{u}{n}), \propto_y = 90 - \frac{180}{\Pi}\arctan(\frac{v}{n}). \tag{8}$$

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:

$$D_x = D_{xu} - D_{tv}, D_y = D_{yv} - D_{tu} \tag{9}$$

where

$$D_{xu} = (x - x_{max}/2)cos(180/\Pi arctan(\frac{u}{n})) \tag{10}$$

$$D_{tv} = (t - t_{max}/2)sin(180/\Pi arctan(\frac{v}{n})) \tag{11}$$

$$D_{yv} = (y - y_{max}/2)cos(180/\Pi arctan(\frac{v}{n})) \tag{12}$$

$$D_{tu} = (t - t_{max}/2)sin(180/\Pi arctan(\frac{u}{n})) \tag{13}$$

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.

4.2.3 Histogram of motion trajectory orientation (HMTO)

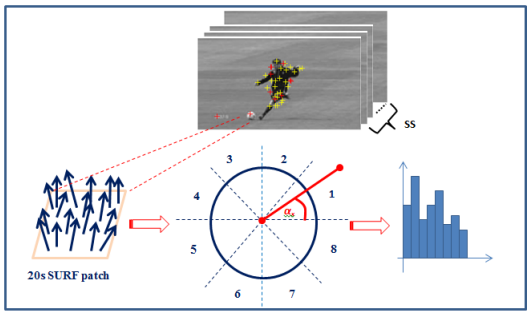

Figure 4. An overview of HMTO extraction.

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[36] or MBH.[32] While other extract spatial information i.e., HOG descriptor.[13] 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 Figure 4).

4.2.4 Motion boundary histogram (MBH)

The motion boundary histogram (MBH) was introduced in[32] 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 motion descriptor for its action recognition discriminative power attested in the state-of-the-art.[18,32]

4.2.5 Spatio-temporal SURF (ST-SURF)

ST-SURF was introduced by.[20] 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.

5. EXPERIMENTS AND RESULTS

5.1 Experimental settings

We start the segmentation process with dense SURF features extraction on a 6×6 sized grid with a temporal step size of $N = 3$ frames so that small motions will not be lost and fast motions will be captured without error. In the case of a still camera, a second clustering of the flow vectors based on the degree of similarity of their magnitudes, angles and closeness is conducted. Thresholds are set 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 the motion of moving interest points. From each clip, we extract local spatio-temporal features as ST-SURF. As described previously, the extracted ST-SURF is a 68D vector (64D SURF, α_x, D_x, α_y, D_y). We also extract square-shaped patches surrounding the detected SURFs. 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, we used motion boundary histogram MBH as a motion descriptor and as a remover of camera motion. MBH_x and MBH_y are 96D histograms. 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^2_χ Kernel.[36]

$$K(v_i, v_j) = \exp(-\sum \frac{1}{A^c} D(v_i^c, v_j^c)), \tag{14}$$

where $D(v_i^c, v_j^c)$ 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.

5.2 Dataset

In this work, experiments are conducted on a large realistic dataset called UCF101.[42] It includes a total number of 101 action classes which are divided into five types: Human-Object Interaction, Body-Motion, Human-Human Interaction, Playing Musical Instruments and Sports. 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[43] and the irrelevant ones are manually removed. All clips have fixed frame rate and resolution of 25 FPS and 320×240 respectively.

5.3 Results and discussion

As described, we use the same settings and evaluation metrics of the state-of-the-art. The accuracy rates reported for the predefined action types are shown in Table 1. For the Sports (87.23%), Playing Musical Instrument (79.4%), Human-Object Interaction (86.07%), Body-Motion Only(85.19%), Human-Human Interaction (88.61%). We can notice that Human-Human Interaction actions achieve the highest accuracy since the spatio-temporal segmentation we introduced in this thesis highlight human bodies, thus the feature extraction is performed in the humans bounding boxes boosts significantly human detection. Performing

sports action achieves a reasonable accuracy of 87.23%, this is due to two factors the first one is the temporal segmentation while the second one is the motion based extraction features. In fact, sports actions show important motion which is very well described in our proposed approaches. Despite that Human-objects and Body motion actions are not based on significant motion, the classification shows satisfactory results. We believe that pixel motion segmentation precision in detecting motion is a good cue to explore human action.

Table 1. Recognition results 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%

We present the results of our approach compared to trajectory and motion based video description approaches. 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, HTMO 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 6.1% achieving 85.3% of accuracy in the challenging realistic big dataset UCF101. Compared with trajectory based descriptors, the proposed approach gives good performances. To represent action-specific scene context, authors in[44] compute local SIFT pyramids on grayscale (P-SIFT) and opponent color keyframes (P-OSIFT) extracted as the central frame of each clip. They improve accuracy by using L1-regularized logistic regression (L1LRS) for stacking classifier outputs 85.7%, 0.2% better than our method. The results given in[45] are lower than ours. In fact, authors provide an extensive empirical evaluation of CNNs on large scale video classification 63.3%. However, in,[46] authors investigate architectures of indiscriminately trained deep Convolutional Networks (ConvNets) for action recognition in video. This method achieves 87.6% which is the best result. This, also, highlights the importance of the classification task investigation, especially in term of deep classification.

Table 2. Some state of the art recognition results over the UCF101 dataset.

Method	Year	Accuracy (%)
Murthy et al.[47]	2013	72.8
Shi et al.[48]	2013	78.9
Wang et al.[49]	2013	85.9
Karaman et al.[44]	2013	85.7
Khuramm et al.[50]	2012	44.5
Karpathy et al.[45]	2014	63.3
Simonyan et al.[46]	2014	87.6

6. CONCLUSION

In this paper, we presented an end-to-end framework for human action recognition in big datasets. 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. 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.

REFERENCES

1. M. Liu, Y. Yan, R. Chen, and H. Wang, "The state-of-the-art research progress on motion segmentation," in *Proceedings of International Conference on Internet Multimedia Computing and Service*, p. 345, ACM, 2014.
2. L. Zappella, X. Lladó, and J. Salvi, "Motion segmentation: A review," in *Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence*, pp. 398–407, IOS Press, 2008.
3. T. Brox, M. Rousson, R. Deriche, and J. Weickert, "Colour, texture, and motion in level set based segmentation and tracking," *Image and Vision Computing* .
4. B. D. Lucas, T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision.," in *IJCAI*, **81**, pp. 674–679, 1981.
5. J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel Corporation* .
6. H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV'06*, pp. 404–417.
7. J. R. Uijlings, A. W. Smeulders, and R. J. Scha, "Real-time visual concept classification," *Multimedia, IEEE Transactions on* .
8. J.-M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *Journal of visual communication and image representation* .
9. M. Jain, H. Jégou, and P. Bouthemy, "Better exploiting motion for better action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13),*, pp. 2555–2562.
10. N. Ikizler-Cinbis and S. Sclaroff, "Object, scene and actions: Combining multiple features for human action recognition," in *Computer Vision–ECCV 2010*, pp. 494–507, Springer, 2010.
11. D. H. Nga and K. Yanai, "A spatio-temporal feature based on triangulation of dense surf," in *IEEE International Conference on Computer Vision Workshops, ICCVW'13*, pp. 420–427.
12. W. Li, Z. Zhang, and Z. Liu, "Expandable data-driven graphical modeling of human actions based on salient postures," *IEEE Transactions on Circuits and Systems for Video Technology* .
13. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'05.*, **1**, pp. 886–893.
14. I. Laptev, "On space-time interest points," *International Journal of Computer Vision* .
15. A. Klaser and M. Marszalek, "A spatio-temporal descriptor based on 3d-gradients," 2008.
16. G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Computer Vision–ECCV'08*, pp. 650–663.
17. P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proceedings of the 15th international conference on Multimedia*, pp. 357–360, ACM, 2007.
18. H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision* , pp. 1–20, 2013.
19. 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 (ICME'10)*, pp. 322–327.
20. S. Megrhi, W. Souidene, and A. Beghdadi, "Spatio-temporal salient feature extraction for perceptual content based video retrieval," in *Colour and Visual Computing Symposium (CVCS'13)*, pp. 1–7.
21. P. Matikainen, M. Hebert, and R. Sukthankar, "Trajectons: Action recognition through the motion analysis of tracked features," in *IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 514–521, 2009.
22. R. Messing, C. Pal, and H. Kautz, "Activity recognition using the velocity histories of tracked keypoints," in *IEEE 12th International Conference on Computer Vision, ICCV'09*, pp. 104–111.
23. O. Oreifej and 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*, pp. 716–723.
24. Y. Song, L. Goncalves, and P. Perona, "Unsupervised learning of human motion models," *Advances in Neural Information Processing Systems* **V.14**, 2003.
25. C. Rao, A. Yilmaz, and M. Shah, "View-invariant representation and recognition of actions," *International Journal of Computer Vision* .

26. N. P. Cuntoor and R. Chellappa, "Epitomic representation of human activities," in *Computer Vision and Pattern Recognition, CVPR'07.*, pp. 1–8.
27. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision* .
28. H. Uemura, S. Ishikawa, and K. Mikolajczyk, "Feature tracking and motion compensation for action recognition.," in *BMVC*, pp. 1–10, 2008.
29. N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," *Image and vision computing* .
30. P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Ninth IEEE International Conference on Computer Vision, ICCV'03.*, pp. 734–741.
31. O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf, "Motion interchange patterns for action recognition in unconstrained videos," in *Computer Vision–ECCV'12*, pp. 256–269.
32. N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Computer Vision–ECCV'06*, pp. 428–441.
33. T.-H. Yu, T.-K. Kim, and R. Cipolla, "Real-time action recognition by spatiotemporal semantic and structural forests," in *Proceedings of the British machine vision conference*, p. 56, 2010.
34. T. Guha and R. K. Ward, "Learning sparse representations for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence* .
35. K. N. Tran, I. A. Kakadiaris, and S. K. Shah, "Modeling motion of body parts for action recognition.," in *BMVC'11*, pp. 1–12.
36. I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *IEEE Conference on Computer Vision and Pattern Recognition. CVPR'08.*, pp. 1–8.
37. G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV'04*, p. 22.
38. A. P. Brandão Lopes, E. Alves do Valle Jr, J. Marques de Almeida, and A. Albuquerque de Araújo, "Action recognition in videos: from motion capture labs to the web," *arXiv preprint arXiv:1006.3506* , 2010.
39. J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence* .
40. R. Brunelli, "Template matching techniques in computer vision,," 2008.
41. D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR'10.*, pp. 2432–2439.
42. K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR* **abs/1212.0402**, 2012.
43. Youtube, "Statistiques@ONLINE," June 2009.
44. S. Karaman, L. Seidenari, A. D. Bagdanov, and A. Del Bimbo, "L1-regularized logistic regression stacking and transductive crf smoothing for action recognition in video," in *ICCV Workshop on Action Recognition with a Large Number of Classes*, 2013.
45. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Conference on Computer Vision and Pattern Recognition*, 2014.
46. K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *arXiv preprint arXiv:1406.2199* , 2014.
47. O. Murthy and R. Goecke, "Ordered trajectories for large scale human action recognition," in *International Conference on Computer Vision Workshops (ICCVW)*, pp. 412–419, IEEE, 2013.
48. F. Shi, R. Laganiere, E. Petriu, and 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.
49. H. Wang and C. Schmid, "Lear-inria submission for the thumos workshop," in *ICCV Workshop on Action Recognition with a Large Number of Classes*, 2013.
50. K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR* **abs/1212.0402**, 2012.