Machine Vision and Applications
https://doi.org/10.1007/s00138-017-0894-7

ORIGINAL PAPER

Spatio-temporal elastic cuboid trajectories for efficient fight recognition using Hough forests

Ismael Serrano[1] · Oscar Deniz[1] · Gloria Bueno[1] · Guillermo Garcia-Hernando[2] · Tae-Kyun Kim[2]

Received: 1 February 2016 / Revised: 4 October 2016 / Accepted: 11 November 2017

Abstract
While action recognition has become an important line of research in computer vision, the recognition of particular events such as aggressive behaviors, or fights, has been relatively less studied. These tasks may be exceedingly useful in some video surveillance scenarios such as psychiatric centers, prisons or even in personal camera smartphones. Their potential usability has caused a surge of interest in developing fight or violence detectors. The key aspect in this case is efficiency, that is, these methods should be computationally very fast. In this paper, spatio-temporal elastic cuboid trajectories are proposed for fight recognition. This method is based on the use of blob movements to create trajectories that capture and model the different motions that are specific to a fight. The proposed method is robust to the specific shapes and positions of the individuals. Additionally, the standard Hough forests classifier is adapted in order to use it with this descriptor. This method is compared to other nine related methods on four datasets. The results show that the proposed method obtains the best accuracy for each dataset and is also computationally efficient.

Keywords Violence recognition · Fight recognition · Descriptor · Blobs · Video sequences · Hough forests

1 Introduction

In recent years, the task of human action recognition from video has been tackled with computer vision and machine learning techniques, see surveys [1–3]. Experimental results have been obtained for recognition of actions such as walking, jogging, pointing or hand waving [4] using STIP features. However, action detection has been denoted comparatively less effort. Violence detection is a task that can be leveraged in real-life applications. While there is a large number of studied datasets for action recognition, important datasets with violent actions (fights) were not available until [5]. The main task of large-scale surveillance systems used in institutions such as prisons, schools and psychiatric care facilities is generating alarms of potentially dangerous situations. Nevertheless, security guards are frequently burdened with the large number of cameras where manual response times are frequently large, resulting in a strong demand for automated alert systems. Also, this type of systems must be very efficient because there is generally a large number of surveillance cameras. Similarly, there is increasing demand for automated rating and tagging systems that can process a large amount of videos uploaded to Web sites. Since smartphones are often used to record beatings, efficient mobile implementations are desired too.

Whereas action recognition techniques focus on many classes, the recognition of a single action may be amenable to more specific algorithms that provide either higher accuracy, better efficiency or both. In the particular case of fight recognition with two classes, the concept of detection is very similar. Then, the only difference is that detection refers to the use of long video sequences, whereas recognition uses trimmed sequences from the original. The latter approach is followed in this work, in accordance with the literature on the topic. The use of this core recognition functionality to build a fight detector is straightforward, and hence, in this work we always refer to the problem as "fight recognition."

In this context, this work hypothesizes that a fight can be described with trajectories of motion areas. A novel descriptor is proposed in order to capture the different parts in motion. This descriptor is called spatio-temporal elastic

✉ Ismael Serrano
ismael.serrano@uclm.es

[1] VISILAB group, University of Castilla-La Mancha, Ciudad Real, Spain

[2] Department of Electrical and Electronic Engineering, University of Imperial College, London, UK

Published online: 07 December 2017

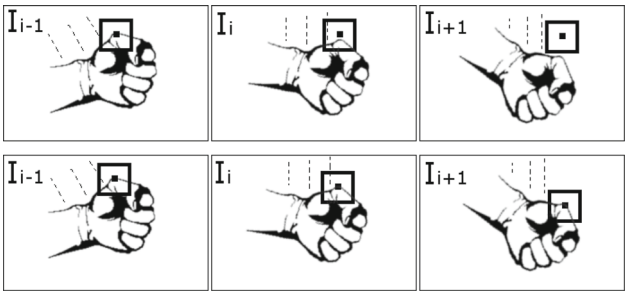

Fig. 1 A fist in motion. Top row: with a classical cuboid. Bottom row: with a STEP trajectory

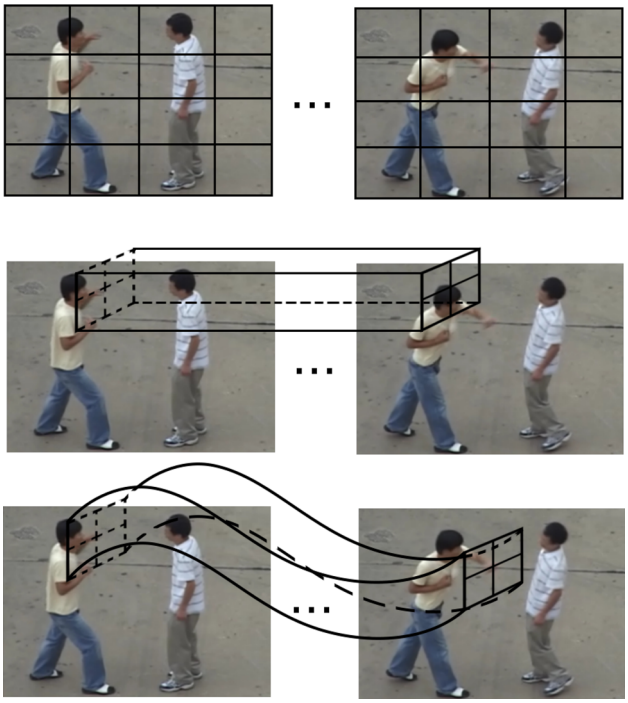

Fig. 2 A fight sequence from UT-Interaction dataset showing a fist. Top row: with dense cuboids. Middle row: with a sparse cuboids (only one around the fist). Bottom row: with a STEC trajectory (also around the fist)

cuboid (STEC) trajectories, see an example in Fig. 1 (bottom) with a fist in motion compared with a classical cuboid (top). See another example in Fig. 2 where a punching action is represented using a STEC trajectory (bottom) compared with a cuboid (middle). Cuboids have been often used to model spatio-temporal changes. STEC trajectories are always centered around tracked parts, whereas the classic cuboids are on a fixed position. It is assumed that background of the videos have to be static or small motion. Thus, the proposed descriptor only focuses on moving parts. Furthermore, in order to represent these STEC trajectories, an extension of Hu et al. [6] method is proposed. These authors propose a novel approach to analyze the topological features of hand postures at multiple scales. We extend this approach in time to represent different parts of an action.

Finally, an adaptation of Hough forests [7–9] is proposed to encode and classify STEC trajectories. In contrast, the classical *bag-of-words* (BoW) [10] model assumes independence between spatio-temporal "words" and does not make use of the rich spatio-temporal relationships inherent in actions. Hough forests leverage this important information. The proposed method and two *state-of-the-art* methods will be compared using BoW or Hough forests approaches in order to verify this claim.

The main contributions of this paper are: the novel STEC trajectories descriptor combined with Hough forests classifier for fight recognition and an extension of Hu et al. [6] method to represent the trajectories.

The paper is organized as follows. Section 2 reviews *state-of-the-art* related methods. Section 3 describes the proposed method. Section 4 provides experimental results. Finally, in Sect. 5 the main conclusions are outlined.

2 Related work

One of the first proposals for violence recognition in video is Nam et al. [11], which propose a method to recognize violent scenes in videos using flame and blood detection and capturing the motion degrees, as well as the characteristic sounds from violent events. Cheng et al. [12] recognized gunshots, explosions and car-braking in audios using a hierarchical approach based on Gaussian mixture and hidden Markov models (HMM). Clarin et al. [13] presented a novel method that uses Kohonen self-organizing maps to search blood and skin areas for each image to detect violence actions where blood appears. Besides, Giannakopoulos et al. [14] proposed a violence detector based on audio features. Zajdel et al. [15] proposed the CASSANDRA system, which extracts motion features related from articulations in video and scream-like cues in audio to search aggressive actions in surveillance videos.

Gong et al. [16] develop a violence detector that uses low-level visual, acoustic features and high-level audio sounds for identifying potential violent action in movies. Chen et al. [17] used binary local motion descriptors (spatio-temporal video cubes) and BoW approach to detect aggressive behaviors. Lin and Wang [18] described a weakly supervised audio violence classifier combined using co-training with motion, explosion and blood to detect violent scenes in movies. Giannakopoulos et al. [19] also proposed a novel method for searching violence actions in movies clustering audio–visual features applying statistics, average motion and motion orientation variance features in video with a K-nearest neighbor classifier to decide whether there are violence actions. Chen et al. [20] proposed a method based in motion, detecting faces

and nearby blood. Hassner et al. [21] recently approached the problem of detecting violence outbreaks in crowds using an optical flow-based method. Proof of the growing interest in the topic is also the MediaEval Affect Task, a competition that goals at searching violence action in color movies [22].

In summary, a significant number of previous *state-of-the-art* methods require audio cues for detecting violence or trust on color areas to detect cues such as blood or skin. In that regard, we note that there is an important number of applications, especially in surveillance, where audio and color features are generally unavailable. In other cases, it is possible and easy to obtain audio features, but audio information may increase false positive rates or decrease true positive rates because there are many violence actions where the audio features may be confused: to push, throw (something), knock down, attack with a knife, block (someone), etc. Moreover, while explosions, blood and running may be very useful cues to detect violence scenarios in action movies, they are unusual in real-world actions. Anyway, violence detection is a very difficult issue, since violence is a subjective concept. Fight detection, on the contrary, is a more specific violence-related topic that may be tackled using similar techniques.

The authors of the local motion patterns (LMP) method [23] claim that it is both informative and efficient for action and violence recognition. Since the number of extracted descriptor vectors varies in each video sequence, this method is based on extracting simple statistics (variance, skewness and kurtosis) from temporal cuboids centered on tracked keypoints. Keypoints are located with a Harris detector. Furthermore, Mohammadi et al. [24] proposed a novel computational framework to classify violence behaviors in various scenes. They focus on capturing the dynamics of pedestrians based on spatio-temporal characteristics of substantial derivatives. They also combine spatial and temporal motion patterns.

More recently, Deniz et al. [25,26] introduced a novel method to recognize fight sequences where blurred areas with specific acceleration patterns are used as the main features. Bermejo et al. [5] demonstrated challenging results using generic action recognition methods to fight detection, obtaining 90% accuracy using MoSIFT features [27]. MoSIFT are powerful features generally applied for generic action recognition. Nonetheless, the computational time cost for extracting features is prohibitively large, taking almost 1 second per frame on a high-performance laptop. The violence flow method (ViF) [21] is a recent method which may be considered representative of dense optical flow-based methods for action recognition. Serrano et al. [28] proposed a novel method considering motion blobs. Simple features (area, perimeter, etc.) are extracted from the largest K blobs and used to discriminate between fights and non-fights.

Tobias et al. [29] utilized Lagrangian measures to detect violent video footage. They focus on Lagrangian coherent structures where they found the directional Lagrangian field a promising feature space that characterizes motion information over a time interval. They propose a local feature set that extends the SIFT algorithm and implements the Lagrangian field to encode the spatio-temporal characteristic of a position.

Gao et al. [30] developed a novel feature extraction method called oriented ViF (OViF) for detecting violence actions, which takes full advantage of the motion magnitude change information in statistical motion orientations based on the aforementioned ViF method [21]. The new feature representation method OViF is proposed considering motion and orientations magnitudes. Feature combination and multiclassifier combination strategies are adopted.

Another aspect related to this work is trajectories that are used to build the proposed spatio-temporal elastic cuboid descriptors. Therefore, it is important to mention some of the most relevant methods based on the concept of trajectories, such as [31–33]. In these methods each trajectory is attached to a particular moving feature. That is, in video deriving from the movement of physical bodies through space, a properly tracks feature (and hence trajectory) that contains useful information about the movement. Wang et al. [34] proposed a robust method that extracts improved dense trajectories (called IDT) for action recognition. This method estimates the camera motion using SURF features and dense optical flow to reject no consistent features. The trajectories are built using the right features. Afterward, HOF and MBH features are extracted for each trajectory. Then, well-known BoW approach is applied to feed a support vector machine (SVM) with linear kernel. They achieve good results on four challenging action datasets.

Finally, some relevant and recent action recognition methods based on deep learning are exposed. Simonyan et al. [35] proposed a deep video classification method for action recognition, which incorporates spatial and temporal recognition streams based on convolutional neural networks (CNN). They used optical flow for training the temporal CNN and achieved significantly better results than training from raw frames. Wang et al. [36] proposed a novel video descriptor, called trajectory-pooled deep-convolutional (TDD) that combines the merits of handcrafted and deep learning features. They used deep architectures to learn discriminative convolutional feature maps, and construct trajectories constrained pooling to aggregate these convolutinal features into effective descriptors. Tran et al. [37] proposed spatio-temporal feature learning using 3-dimensional convolutional neural networks (3D-CNN) for action recognition, called C3D. The authors argue that 3D-CNN is able to model temporal information thanks to the 3D pooling and 3D convolution

layers. The method achieves 85.2% action recognition accuracy on the challenging UCF101 dataset.

3 Proposed method

The proposed method consists of two independent stages: the spatio-temporal elastic cuboid (STEC) trajectories (descriptor) and Hough forests (classifier). The following subsections describe both stages.

3.1 Spatio-Temporal Elastic Cuboid trajectories

In the following method, it is assumed that background of the videos is static.

In this work, it is hypothesized that a fight can be described with the position, size, rotation and shape of motion areas. Figure 2 shows a simple example of a STEC trajectory that tracks a part in motion (the fist). *State-of-the-art* methods [9,38–41], etc.) that extract spatio-temporal features use cuboids from a grid (Fig. 2 top) or from salient features (Fig. 2 middle) in videos. In this work we use "elastic" cuboids (or STEC trajectories) (Fig. 2 bottom) that track different parts in motion and then extract features. The "elastic" concept is used to differentiate them from the classic cuboids that have a fixed position throughout neighboring frames.

Note that pixel tracking approaches (generally based on optical flow) may not be appropriate in our context for two reasons. Firstly, the assumption of "immediate neighborhood of (x, y) is translated some small distance $(\mathrm{d}x, \mathrm{d}y)$ during the interval." Therefore, aggressive actions (i.e., with very fast motion) should be subject to failures due to the previous assumption. Secondly, the processing time on cameras, on these aggressive actions, usually produces blur areas that affect pixel-level tracking. On the other hand, these approaches are computationally slower, which may preclude implementations in resource-limited hardware (like smartphones). For all these reasons, we decided to focus on region tracking over consecutive frames. In a sense, the approach here is not to attempt fine-detail tracking but instead focus on coarser changer.

The process to obtain the STEC trajectories is summarized in the following:

1. A binarized image is obtained containing movements between consecutive frames. A standard binarization method is applied using an adaptive threshold. An image with motion blobs is obtained for every two consecutive frames.
2. The best K blobs are selected from the binarized images.
3. A body part in motion (for example, the hand in a punch) is modeled by a trajectory of motion blobs (STEC). Here, it is necessary to link the motion blobs in the previous image with those in the current image. A method is proposed to associate two consecutive blobs using distance, area and shape. With these linked blobs a STEC trajectory is built.
4. Characterize STEC trajectories. To model a STEC trajectory, relative sizes, positions, orientations and shapes are obtained.

In the following, these steps are described in detail:

Firstly, a short sequence $S(s)$ of gray images is denoted as:

$$S(s) = I_1, I_2, \ldots, I_t, \ldots, I_T \tag{1}$$

where $s \in \mathbb{Z}$ (number of sequences), with size $N \times M$. T is the number of frames, and N and M are the number of rows and columns for $S(s)$ sequence.

Let I_{t-1} and I_t be two consecutive frames in the sequence. The absolute difference between consecutive images is then computed as:

$$E_t = |I_{t-1} - I_t| \tag{2}$$

Then, E is binarized using a traditional binarization method:

$$F_t = \begin{cases} 1, & \text{if } E_t > 255 \cdot H_t, \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where H_t is an adaptive threshold used to binarize each E_t image, $0 < H_t < 1$ and $H_t \in \mathbb{R}$, calculated with the Otsu method. These binarized images contain the areas with motion.

The second step is to locate and select a set of blobs in each image F_t. The K largest blobs are selected for each binarized image. The blobs are represented as a function $B_{b,t} = b$ where $b \in \{0, 1, \ldots, J\}$ is the unique index for each blob in the image, and J is the number of blobs in the image F_t. Each blob $B_{b,t}$ is obtained as an image that contains a set of adjacent points, neighborhood pixels and where their values are 1 from F_t. These blobs are calculated from each binarized image. Afterward, the K largest blobs are selected using their blob areas ($A_{b,t}$) that are calculated counting the pixel number inside each blob. $A_{1,t}, \ldots, A_{b,t}, \ldots, A_{J,t}$ are the number of adjacent pixels (or area) of each blob, respectively, and $A_{1,t} + \cdots + A_{b,t} + \cdots + A_{J,t} \leq N \cdot M$

Here it is assumed that the blobs with largest areas are the most representative in each frame. However, when there is little or no movement in the frame, some of these K largest blobs may not be representative. Therefore, the m_t (minimum threshold area) is defined to reject blobs with little area as:

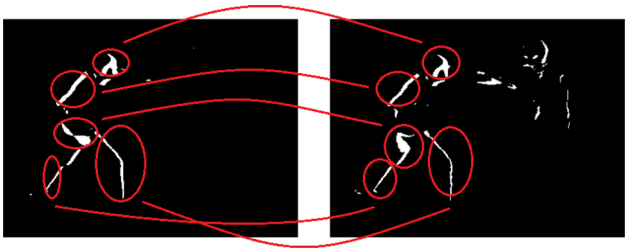

Fig. 3 A punching sequence with the best five blobs marked in two consecutive frames. These five blobs have been also linked in this pair of frames

$$m_{\mathrm{t}} = \begin{cases} \frac{1}{J}\sum_{b=1}^{J} A_{b,t}, & \text{if } \frac{1}{J}\sum_{b=1}^{J} A_{b,t} > M_{\mathrm{a}} \cdot N \cdot M \\ M_{\mathrm{a}} \cdot N \cdot M, & \text{otherwise} \end{cases} \tag{4}$$

where M_{a} (minimum areas) is a parameter to set, $0 > M_{\mathrm{a}} > 1$ and $M_{\mathrm{a}} \in \mathbb{R}$. Finally, if $A_{b,t} < m_t$, then the blob B_b is rejected. Note that K is thus a maximum of blobs, and if there is little motion fewer blobs will be selected.

The third step is to link a selected blob in the current frame with another one in the next frame. Let us use a sequence where K blobs have been selected (example, see the sequence in Fig. 3 using 5 marked blobs). For each blob in frame t we need to find where the next blob is in $t+1$. For this purpose, a match measure will be defined for a blob in t and every blob in $t+1$. This measure is calculated based on the relative areas, distances and shapes of the two blobs. Let us use $B_{b,t}$ for $B_{b,t}$; this measure is defined as:

$$\varphi(B_{b,t};\, B_{b',t+1}) = \frac{1}{\mathrm{area}(B_{b,t},\, B_{b',t+1}) \cdot \mathrm{dist}(B_{b,t},\, B_{b',t+1}) \cdot \mathrm{shape}(B_{b,t},\, B_{b',t+1})} \tag{5}$$

where $\mathrm{area}(B_{b,t},\, B_{b',t+1})$ represents the absolute difference of two area blobs and can be rewritten as: $\mathrm{area}(B_{b,t},\, B_{b',t+1}) = \mid A_{b,t+1} - A_{b',t} \mid$. The term $\mathrm{dist}(B_{b,t+1},\, B_{b',t})$ represents the distance between two points. The centroids of the blobs are used to represent each blob position and calculate their distances. This part of the equation can be rewritten as:

$$\mathrm{dist}(B_{b,t},\, B_{b',t+1}) = \sqrt{(CX_{b,t} - CX_{b',t+1})^2 + (CY_{b,t} - CY_{b',t+1})^2}, \tag{6}$$

where $CX_{b,t}$ and $CY_{b,t}$ are the centroids of a blob that is defined as:

$$CX_{b,t} = \frac{1}{m_b}\sum_{B_{b,t}=b} x,\; CY_{b,t} = \frac{1}{m_b}\sum_{B_{b,t}=b} y \tag{7}$$

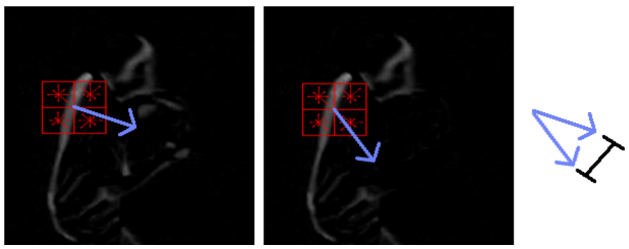

Fig. 4 Left and middle: 2 consecutive difference images from a punching sequence where the HOG method is applied on the blob center; the blue lines are the predominant direction vectors (amplified). Right: these blue line vectors are subtracted and the magnitude is calculated, the black line (color figure online)

The last part of the equation is the term $\mathrm{shape}(B_{b,t},\, B_{b',t+1})$ that provides a shape difference between the two blobs. The blob $B_{b,t}$ shape is estimated using histogram of oriented gradients (HOG) method that is applied on the difference image region for blob $B_{b,t}$. An example is shown in Fig. 4, where the HOG method is applied on the blob center to estimate its shape (as the predominant angle). The vector of maximum gradient is used ($\mathrm{HOG}_{\max}(B_{b,t})$). Then, this part of the equation can be rewritten as:

$$\mathrm{shape}(B_{b,t},\, B_{b',t+1}) = \mid\mid \mathrm{HOG}_{\max}(B_{b,t}) - \mathrm{HOG}_{\max}(B_{b,t+1}) \mid\mid \tag{8}$$

In summary, these two HOG vectors have been obtained and subtracted, and the magnitude of this result vector is calculated.

Now each pair of blobs should be linked with the following maximization function where the best blob b' is selected:

$$\max_{b'} \left(\varphi(B_{b,t};\, B_{b',t+1})\right) \tag{9}$$

Each pair of blobs is linked in time t. This process is repeated each time step to build the trajectories. Parameter L (length) limits the number of blobs paired along a trajectory, for example see Fig. 5 with $L = 5$ blobs. In order to reduce mismatches, i.e., pairs of blobs that are far away and very

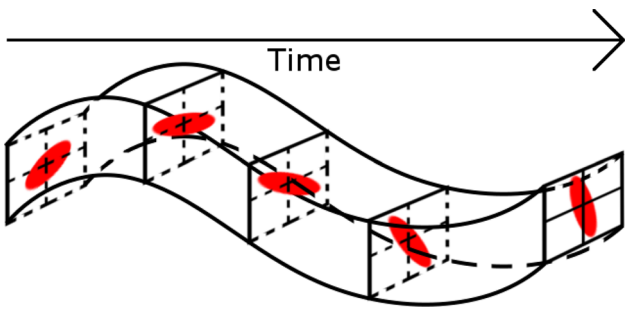

Fig. 5 A 5-length STEC trajectory where the blobs have been linked in time

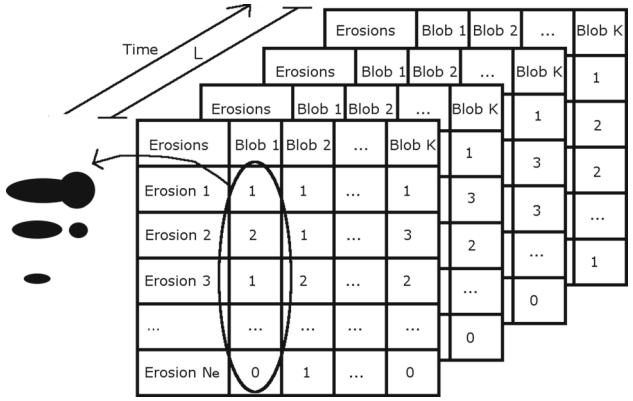

Fig. 6 Left: the first blob with three erosions. Right: the matrix result for K trajectories of L blobs using the extended erosion method. The column with the ellipse is the result for the first blob that counts the number of regions for each erosion step

different, parameter M_t (minimum trajectories) is introduced to reject trajectories, where M_t can be chosen between $0 > M_t > 1$ and $M_t \in \mathbb{R}$. Finally, if $\text{area}(B_{b,t}, B_{b',t+1}) > M_t \cdot N \cdot M$ or $\text{dist}(B_{b,t}, B_{b',t+1}) > d_{\max} \cdot M_t$, then the current trajectory is rejected, where $d_{\max} = \sqrt{N^2 + M^2}$ (diagonal distance of one frame).

The last step is how to describe each trajectory. In order to build a robust descriptor, relative features between pairs of consecutive blobs in the trajectory are calculated. Different features of the blobs are used to encode the position, size, orientation and shape. It is proposed to use 9 relative measures of each blob ($9(L-1)$ features), the length trajectory (1 feature) and a set of features that encodes the shape in space–time (x, y, t) ($N_e \cdot K \cdot L$ features). The first 9 relative measures are centroid (x and y), area, distance, perimeter, major and minor axes, orientation and number of other blobs. These relative measures are calculated by subtracting the values at t and $t+1$. The centroid, area and distance were already defined. To estimate the perimeter ($P_{b,t}$) the Sobel operator is used to detect the edges on $B_{b,t}$. In summary, the proposed method obtains $L \cdot (N_e \cdot K + 9) - 8$ features per trajectory.

The features that can model the shape of the trajectories in space–time ($L \cdot N_e$ features) come from an extension of [6]. In [6] the authors proposed a novel approach to analyze the topological features of hand postures at multiple scales. They computed the convex hull on the hand region and considered the complementary space of the hand as holes. Then, they applied multiple morphological erosion operations (N_e) over these holes and counted the number of regions that are formed. In this paper, an extension of this approach is proposed. This method is applied for each blob on the trajectory. Then, the sequence shape in space–time can be modeled. An example using the multiscale erosion method in space–time is shown in Fig. 6 for K trajectories.

3.2 Hough forests

Hough forests [8,9] consist of a set of random trees [42] that are trained to learn a mapping from densely sampled D-dimensional feature *cuboids* to their corresponding votes in a Hough space $\mathbb{H} \subseteq \mathbb{R}^H$. The Hough space encodes the hypothesis for an object/action position in scale(time)–space and class. The term *cuboid* below is used in a generalized sense to represent a local image patch ($D = 2$) or video space-temporal neighborhood ($D = 3$) depending on the task.

Each tree τ in Hough forests $\tau = T_{\text{tree}}$ is constructed from a set of feature *cuboids* $P_i = (F_i, c_i, d_i)$ that are randomly sampled from the image or video sequence where F_i are the extracted features from a *cuboid* of fixed size (D) in $\mathbb{R}^D$, c_i is the class label for the sample, and d_i is a displacement vector from the cuboid pointing toward the spatio-temporal center of the action. The negative classes have $d_i = 0$. In [8,9], cuboids are used with fixed dimensions 16×16 and $16 \times 16 \times 5$ for images and videos, respectively. Then, for each *cuboid* the grayscale intensity, absolute value of x, y and time derivatives, absolute value of optical flow in x and y are obtained. In this work, Hough forests are trained with the STEC trajectories.

Each leaf node L stores the probability of the cuboids belonging to the object class $\varphi(c \mid L)$, estimated as the proportion of cuboids per class label reaching the leaf after training, and $D_c^L = \{d_i\}_{c_i = c'}$ the cuboids respective displacement vectors. In this work, each non-leaf node is modified to assign a binary test from a set of input vector features (F). The binary test is now defined by a composition of two features values $(p, q) \in \mathbb{R}^D$ with some offset O_s. The binary test (b) on a non-leaf node is redefined as:

$$b_{p,q,O_s}(F) = \begin{cases} 1, & \text{if } F(p) < F(q) \cdot O_s \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

The next steps of the Hough forests method are constructed according to [9]. The Hough forests output for a sequence (set of features) is an image with their corresponding votes in Hough space $\mathbb{H} \subseteq \mathbb{R}^H$. Besides, the maxima class probability can be searched by applying a Parzen estimator with a Gaussian kernel. The accumulation of the probabilities is made with the summation criteria to make the final decision.

The Hough forests are constructed using whole STEC trajectories that were extracted from the training set. This training step is computationally slow since the number of operations is exponentially related to depth of trees. Afterward, a set of decision trees have been created. When a testing STEC trajectory is tested on the created Hough forests, it gives a class probability and prediction in a spatio-temporal

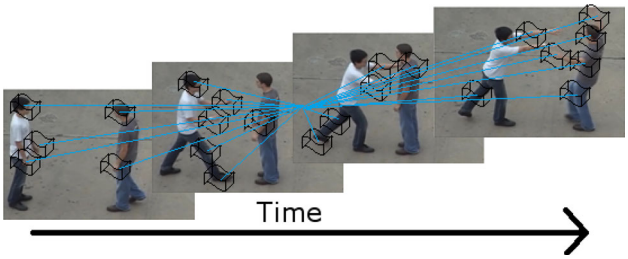

Fig. 7 A pushing sequence with some STEC trajectories represented. Each STEC trajectory points to the action center

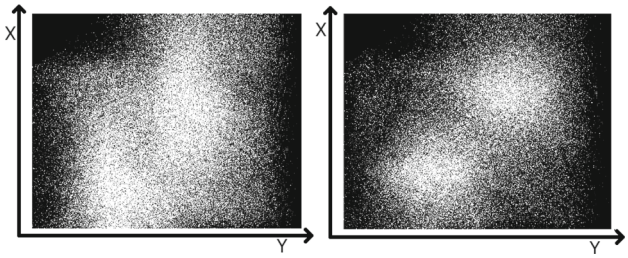

Fig. 8 Sum space votes in time from a fight and non-fight *Hockey* sequence, respectively. Left: the fight space votes. Right: the non-fight space votes

position. The testing step is computationally very efficient because the complexity is logarithmic.

The main reason to use Hough forests in this work is: this classifier does not need a previous BoW to cluster (or encode) and represent the trajectories for each sequence. Each sequence has a different number of trajectories that depend on the amount of motion. Then, no matter what classifier is to be used, a previous clustering is needed. On the other hand, under a BoW framework the sequencing of the STEC trajectories would be lost, and this aspect is very important to model a fight action. Hough forests use all trajectories (features) independently without losing temporal information, see an example in Fig. 7 where each STEC trajectory gives spatio-temporal information about the action and its center.

The Hough forests classifier creates a Hough voting space when a new sequence (set of features) is tested. Each feature produces a vote in space–time and class. To see an example on a fight *Hockey* sequence, the mean of the space votes in time is represented in Fig. 8. It is now possible to know where the fight/non-fight is. Besides, the accumulation of votes is calculated for each time step (see Fig. 9), so it is also possible to know when the fight/non-fight occurs.

4 Experiments

The proposed method is assessed using four different datasets and compared to nine related methods. In order to evaluate the performance of the proposed descriptor and classifier,

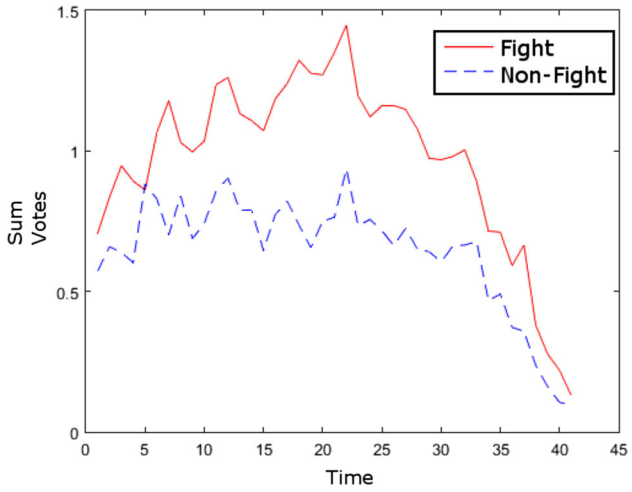

Fig. 9 Sum votes in space for each time step. Using the same fight *Hockey* sequence

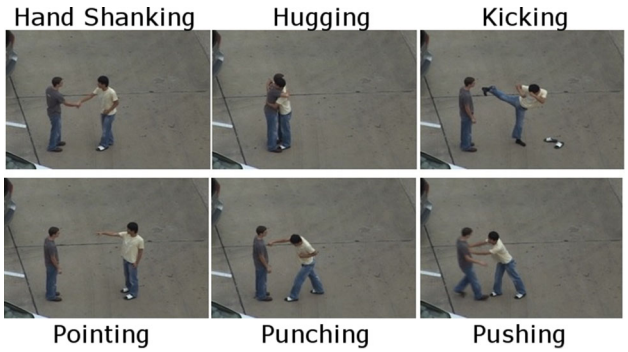

Fig. 10 UT-Interactions dataset (set 1) with the six classes

a sparse set of STEC trajectories is obtained from each sequence. Then, using this sparse set of features, a Hough forests classifier is trained and tested.

4.1 Datasets

The first dataset [43] is *UT-Interaction* that contains videos of six classes of human–human interactions: *shake hands*, *hug*, *kick*, *point*, *punch* and *push*. There are 20 video sequences in total. Each video contains at least one execution of the action, providing 8 execution of human activities per video on average. The dataset is divided into two sets. Set 1 is recorded in a parking lot with a stationary background and set 2 on a lawn with slight background movement and camera jitter. The authors of this dataset also give the segmented actions. In each set, there are 10 segmented sequences per class, in total 60. These sequences are used and clustered in two classes: fights and non-fights. The fight classes are kick, *punch* and *push actions*, and non-fight classes are the other three classes. These two sets are called "*UT1*" and "*UT2*." See an example of this dataset (set 1) in Fig. 10.

Fig. 11 Sample fight videos from the action movie (above) dataset and the *Hockey* (below) dataset

The work [5] introduced two datasets explicitly designed for assessing fight detection. The first dataset ("*Movies*") introduced in [5] consists of 200 video clips in which fights are extracted from action movies (see Fig. 11 above). Each clip is limited to 50 frames and resolution lowered to 320 × 240. These videos depict a wider variety of scenes and are captured at different resolutions. The non-fight videos are extracted from public action recognition datasets. The second dataset ("*Hockey*") consists of 1000 clips at a resolution of 720 × 576 pixels, divided into two groups, 500 fights (see Fig. 11 below) and 500 non-fights, extracted from hockey games of the National Hockey League (NHL).

4.2 Method parameters

In order to obtain an appropriate set of parameters, a grid search optimization method was used. This simple optimization method is performed on the training/validation part of the *UT1* dataset (80 and 10% of the dataset, respectively). The value being optimized is the average classification accuracy with 10-fold cross-validation. Besides, to reduce variance, three runs are separately performed and the average of accuracy is considered. The following parameters of the proposed algorithm are optimized: the maximum number of motion blobs per frame (K), minimum areas threshold (M_a), minimum trajectories threshold to link blobs (M_t), trajectory length (L) and number of erosions (N_e). For the optimization, the parameter space is divided into a rough grid with manually selected points.

The optimization converged to the following set of parameters: the maximum number of motion blobs per frame $K = 10$; minimum areas $M_a = 0.0001$; minimum trajectories to link blobs $M_t = 0.2$; trajectory length $L = 4$; and number of erosions $N_e = 6$ for *UT1* dataset. After that, it is not needed to re-tune the parameters for the remaining datasets. The parameters obtained from *UT1* are used for the other datasets.

The proposed method also estimates the shape between blobs with the HOG algorithm, and it has been used with these standard values (as [44]): the block size is 2 × 2, cell size is 8 × 8, overlap is half the block size, number of bins is 9 and non-signed orientation is used.

Accuracy is defined as the measure used to compare and obtain results. Accuracy is calculated as the number of the true positives (TP) plus true negatives (TN) divided by the total number of samples. It is a good measure because number of video examples is equilibrated.

4.3 Experimental results

In all subsequent experiments, the parameters used were those obtained as discussed in Sect. 4.2. The Hough forests classifier was trained with 10 trees, depth 15 and 100 splits per node (as [8]).

In order to compare the proposed method, the following recent methods were used with one or more datasets. The methods [24,30,37] which were explained in Sect. 2 use one or more datasets.

Seven related methods have been tested for each dataset. The first three methods are: the violence flow method (ViF) [21], the local motion patterns method (LMP) [23] and the method [28] that were mentioned on Sect. 2. The LMP method needs fixed-size histograms to extract the features. The best results are obtained using 6 bins. These three methods have been used with Random Forests classifier because it is more akin to the classifier used and usually gets the best accuracy. The last four methods are: space–time interest points (STIP) [4] and improved dense trajectories (IDT) [34] that were also mentioned on Sect. 2, following two approaches. The first approach is carried out following the identical setting to [34], and it is repeated for STIP features. The STIP and IDT features are extracted for each dataset. To encode the features, BoW is used with a 4000-bin codebook. Afterward, a SVM with linear kernel is trained and tested. The second approach is carried out using the STIP and IDT features with Hough forests classifier using the mentioned parameters.

Table 1 Ten related methods compared with the proposed method (STEC) on some or all datasets

	*UT*1 (%)	*UT*2 (%)	Movies (%)	Hockey (%)
[24]	–	–	96.9	–
[37]	–	–	93.6	87.4
[30]	–	–	87.5	–
ViF + RF	81.7	78.3	88.9	82.4
LMP + RF	60	53.3	92	77.7
[28] + RF	86.7	91.7	97.8	82.4
STIP + BoW	71.7	56.6	72	75.4
STIP + HF	98.4	98.3	94.5	**89.2**
IDT + BoW	61.7	78.3	74.5	57.3
IDT + HF	80	95.5	84	80
STEC + BoW	58.3	55	70	65.3
STEC + HF	**98.6**	**99.5**	**98**	82.6

Bold values show the best results for the dataset

Table 2 Feature extraction times. Average times measured on *UT1* dataset, on an Intel Xeon computer with 2 processors at 2.90 Ghz

Method	Features/sequence	Msecs/frame
ViF	96000	454.5
LMP	10368	151.6
[28]	2940	26.5
STIP	–	163.2
IDT	–	996.1
STEC	–	61.8

These ten related methods are now compared with the proposed method (STEC) on some or all datasets (the IDT approach is also applied to the STEC trajectories) (see Table 1). The proposed method (STEC + HF) achieves the best results in three of the four datasets. It can be seen that the Hough forests approach achieves better results to encode and classify spatio-temporal features (as STIP, IDT and STEC) than BoW + SVM approach. Note that the results in the first column of Table 1 were obtained with the test part of *UT1* dataset, which is disjunct from the validation part mentioned in Sect. 4.2.

Table 2 shows the number of features used for classification and the computational cost (for feature extraction). These results show that the proposed method is not the fastest one, although the difference is very small compared to the fastest [28]. However, the accuracy is significantly better. Note that the STEC, STIP and IDT features do not give a fixed number of features per sequence, because these methods obtain features where motion occurs. Therefore, they depend on the input sequence. It can be seen that the proposed method achieves a right trade-off between accuracy and computational time.

The proposed method is a simple yet efficient tool for violence or fight recognition. The accuracy has been compared to the *state-of-the-art* methods with four datasets containing violence actions. The accuracy was significantly higher in the three of the four compared datasets. The boost in this accuracy can be explained by the ability of STEC trajectories to capture the different motion parts in space–time. Each STEC trajectory models a part of the action, and it points to the action center in space–time. Also the Hough forests classifier does not lose temporal transition information that is considered useful. Another positive aspect is the low extraction times, which can be explained because a sparse sampling is used to represent the actions. This is in contrast with most *state-of-the-art* methods, which need a dense sampling to represent the actions. Again, efficiency in this task is paramount.

5 Conclusions

The spatio-temporal elastic cuboid (STEC) trajectories descriptor is a novel proposed descriptor to model the different parts in motion in fight or non-fight sequences. These STEC trajectories capture the relative position, size, orientation and shape of the movements. The descriptor is robust to scale, orientation and position changes due to the use of relative measures between consecutive frames. Furthermore, this method does not need a person detector to find the actors as other *state-of-the-art* methods because the whole sequence is used. Also, the method does not use trackers that are generally computationally expensive and may require manual initialization. The proposed method does not use a BoW method to cluster the trajectories (features), for the adapted Hough forests classifier is used to keep the temporal order of the action.

The proposed method has been compared with ten other methods in some or all of the four datasets. Experiments show that the method is better than the related methods in the three of the four datasets. Although it is not the fastest one, the trade-off between computational time and accuracy is clearly better. This method gives an accuracy between 98 and 99.5% in *UT1*, *UT2* and *Movies* datasets. The accuracy in the *Hockey* dataset is 82.5%.

Note that pixel tracking approaches (generally based on optical flow) may not be appropriate in our context for two reasons. Firstly, aggressive actions (i.e., with very fast motion) should be subject to failures due to the previous assumption. Secondly, the processing time on cameras, on these aggressive actions, usually produces blur areas that affect pixel-level tracking. On the other hand, these approaches are computationally slower, which may preclude implementations in resource-limited hardware. In a sense,

the approach here is not to attempt fine-detail tracking but instead focus on coarser changer.

The Hough forests approach achieves better results to encode and classify spatio-temporal features (as STIP, IDT and STEC) than the BoW + SVM approach.

While other methods tackle a more general problem (such as action recognition) or resort to computationally intensive optical flow computations, the proposed method opens up the possibility of practical implementations. There is growing interest in the private video surveillance sector in deploying efficient methods for violence detection in prisons and other similar scenarios.

Future work will seek to improve accuracy by using additional features to detect the violent areas. In general, STEC trajectories method provides an excellent balance between high detection accuracy and feature extraction times, and it is expected than an implementation of this method on a GPU would give an extra significant speed-up.

Acknowledgements This work has been partially supported by Project TIN2011-24367 from Spain's Ministry of Economy and Competitiveness.

References

1. Poppe, R.: A survey on vision-based human action recognition. Image Vis. Comput. **28**(6), 976–990 (2010)
2. Turaga, P., Chellappa, R., Subrahmanian, V., Udrea, O.: Machine recognition of human activities: a survey. Circuits Syst. Video Technol. IEEE Trans. **18**(11), 1473–1488 (2008)
3. Shian-Ru, K., Hoang Le Uyen, T., Yong-Jin, L., Jenq-Neng, H., Jang-Hee, Y., et al.: A review on video-based human activity recognition. Computers **2**(2), 88–131 (2013)
4. Laptev, I., Lindeberg, T.: Space-time interest points. In: Proceedings of International Conference on Computer Vision, pp. 432–439. (2003)
5. Bermejo, E., Deniz, O., Bueno, G., Sukthankar, R.: Violence detection in video using computer vision techniques. In: 14th International Congress on Computer Analysis of Images and Patterns, pp. 332–339. (2011)
6. Hu, K., Yin, L.: Multi-scale topological features for hand posture representation and analysis. In: IEEE International Conference on Computer Vision (ICCV), 2013, pp. 1928–1935. (2013)
7. Waltisberg, D., Yao, A., Gall, J., Van Gool, L.: Variations of a Hough-voting action recognition system. In: Ünay, D., Çataltepe, Z., Aksoy, S. (eds.) Recognizing Patterns in Signals, Speech, Images and Videos, pp. 306–312. Springer, Berlin (2010)
8. Gall, J., Yao, A., Razavi, N., Van Gool, L., Lempitsky, V.: Hough forests for object detection, tracking, and action recognition. Pattern Anal. Mach. Intell. IEEE Trans. **33**(11), 2188–2202 (2011)
9. Yao, A., Gall, J., Van Gool, L.: A Hough transform-based voting framework for action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 2061–2068. (2010)
10. Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. Int. J. Comput. Vis. **79**(3), 299–318 (2008)
11. Nam, J., Alghoniemy, M., Tewfik, A.: Audio-visual content-based violent scene characterization. In: Proceedings of ICIP, pp. 353–357. (1998)
12. Cheng, W., Chu, W., Wu, J.L.: Semantic context detection based on hierarchical audio models. In: Proceedings of the ACM SIGMM Workshop on Multimedia Information Retrieval, New York, pp. 109–115. (2003)
13. Clarin, C., Dionisio, J., Echavez, M., Naval, P.: Dove: detection of movie violence using motion intensity analysis on skin and blood. PCSC **6**, 150–156 (2005)
14. Giannakopoulos, T., Kosmopoulos, D., Aristidou, A., Theodoridis, S.: Violence content classification using audio features. In: Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 3955, pp. 502–507. (2006)
15. Zajdel, W., Krijnders, J., Andringa, T., Gavrila, D.: CASSANDRA: audio-video sensor fusion for aggression detection. In: IEEE Conference on Advanced Video and Signal Based Surveillance, 2007, pp. 200–205. (2007)
16. Gong, Y., Wang, W., Jiang, S., Huang, Q., Gao, W.: Detecting violent scenes in movies by auditory and visual cues. In: Proceedings of the 9th Pacific Rim Conference on Multimedia, pp. 317–326. Springer, Berlin (2008)
17. Chen, D., Wactlar, H., Chen, M., Gao, C., Bharucha, A., Hauptmann, A.: Recognition of aggressive human behavior using binary local motion descriptors. In: Engineering in Medicine and Biology Society, 2008. (20–25 2008) pp. 5238–5241 (2008)
18. Lin, J., Wang, W.: Weakly-supervised violence detection in movies with audio and video based co-training. In: Proceedings of the 10th Pacific Rim Conference on Multimedia, pp. 930–935. Springer, Berlin (2009)
19. Giannakopoulos, T., Makris, A., Kosmopoulos, D., Perantonis, S., Theodoridis, S.: Audio-visual fusion for detecting violent scenes in videos. In: 6th Hellenic Conference on AI, SETN 2010, Athens, Greece, May 4–7, 2010. Proceedings, pp. 91–100. Springer, London (2010)
20. Chen, L., Su, C., Hsu, H.: Violent scene detection in movies. IJPRAI **25**(8), 1161–1172 (2011)
21. Hassner, T., Itcher, Y., Kliper-Gross, O.: Violent flows: real-time detection of violent crowd behavior. In: 3rd IEEE International Workshop on Socially Intelligent Surveillance and Monitoring (SISM) at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2012)
22. Demarty, C., Penet, C., Gravier, G., Soleymani, M.: MediaEval 2012 affect task: violent scenes detection in Hollywood movies. In: MediaEval 2012 Workshop, Pisa (2012)
23. Ward, R.K., Guha, T.: Learning sparse representations for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **34**(8), 1576–1588 (2012)
24. Mohammadi, S., Kiani, H., Perina, A., Murino, V.: Violence detection in crowded scenes using substantial derivative. In: International Conference on Advanced Video and Signal-based Surveillance, AVSS, (2015)
25. Deniz, O., Serrano, I., Bueno, G., Kim, T.K.: Fast violence detection in video. In: The 9th International Conference on Computer Vision Theory and Applications (VISAPP), (2014)
26. Serrano, I., Déniz, O., Bueno, G.: Visilab at MediaEval 2013: fight detection. In: MediaEval 2013, vol. 1043. MediaEval Bechmark (2013)
27. Chen, M., Mummert, L., Pillai, P., Hauptmann, A., Sukthankar, R.: Exploiting multi-level parallelism for low-latency activity recognition in streaming video. In: MMSys '10: Proceedings of the First Annual ACM SIGMM Conference on Multimedia Systems, New York, pp. 1–12. (2010)
28. Serrano, I., Deniz, O., Bueno, G., Kim, T.K.: Fast fight detection. PLoS ONE **10**(4), e0120448 (2015)

29. Tobias, S., Volker, E., Thomas, S.: A local feature based on Lagrangian measures for violent video classification. In: 6th International Conference on Imaging for Crime Prevention and Detection, IET (2015)
30. Gao, Y., Liu, H., Sun, X., Wang, C., Liu, Y.: Violence detection using oriented violent flows. Image Vis. Comput. **48**, 37–41 (2016)
31. Matikainen, P., Hebert, M., Sukthankar, R.: Trajectons: Action recognition through the motion analysis of tracked features. In: IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), 2009, pp. 514–521. (2009)
32. Messing, R., Pal, C., Kautz, H.: Activity recognition using the velocity histories of tracked keypoints. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 104–111. (2009)
33. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3169–3176. (2011)
34. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: 2013 IEEE International Conference on Computer Vision (ICCV), pp. 3551–3558. (2013)
35. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., Weinberger, K. Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 568–576. Curran Associates, Inc. (2014)
36. Wang, L., Qiao, Y., Tang, X.: Action recognition with trajectory-pooled deep-convolutional descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4305–4314. (2015)
37. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 4489–4497. (2015)
38. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005, pp. 65–72. (2005)
39. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3361–3368. (2011)
40. Taylor, G.W., Fergus, R., LeCun, Y., Bregler, C.: Convolutional learning of spatio-temporal features. In: Computer Vision–ECCV 2010, pp. 140–153. Springer, Berlin (2010)
41. Willems, G., Tuytelaars, T., Van Gool, L.: An efficient dense and scale-invariant spatio-temporal interest point detector. In: Computer Vision–ECCV 2008, pp. 650–663. Springer, Berlin (2008)
42. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
43. Ryoo, M.S., Aggarwal, J.K.: Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: IEEE International Conference on Computer Vision (ICCV), (2009)
44. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. (2008)

Dr. Ismael Serrano His research interests are mainly focused on computer vision and machine learning, especially on deep learning. He is the author of more than 15 papers in journals and conferences. He has published two books on OpenCV. He has served as researcher collaborator at Imperial College London (UK) and Leica Biosystems (Ireland). He received the degree in computer science in 2012 from the University of Castilla-La Mancha. He scored the highest mark in his final degree project about person detection. He obtained his PhD with "Cum Laude" mention from University of Castilla-La Mancha, Spain, in 2016 about fight detection in video using computer vision and machine learning techniques.

Dr. Oscar Deniz His research interests are mainly focused on computer vision and pattern recognition. He is the author of more than 50 refereed papers in journals and conferences. He has published two books on OpenCV. He has served as visiting researcher at Carnegie Mellon University (USA), Imperial College London (UK) and Leica Biosystems (Ireland). Currently, he works as an associate professor at UCLM and contributes to VISILAB. He is a Senior Member of IEEE and is affiliated with the AAAI, SIANI, CEA-IFAC, AEPIA, AERFAI-IAPR and the Computer Vision Foundation. He serves as an Academic Editor of Journal PLoS ONE and Associate Editor of IEEE Consumer Electronics. He is the coordinator of European Project "Eyes of Things", an Innovation Action to be funded within the H2020 programme. He is a Reviewer/Technical Expert for EU programmes such as Eurostars.

Dr. Gloria Bueno Is a Lecturer and Principal Research at School of Engineering in UCLM, at Ciudad Real, Spain, since 2002. She holds a PhD in Machine Vision obtained at Coventry University in 1998. She has carried on her research activities at different research centres, such as Centro de Estudios e Investigaciones Técnicas de Guipuzkoa – San Sebastián (E), Centre National de la Recherche Scientifique, Hôpitaux Civil & Telecommunication School, Louis Pasteur University, Strasbourg (FR) and Gilbert Gilkes & Gordon Technology, Kendal (UK). She is leading different national and European research projects in biomedical image processing. Her current interests are in signal and image processing, modelling and artificial intelligence.

Guillermo Garcia-Hernando Is currently a PhD student at the Computer Vision and Learning Lab at Imperial College London, UK. He received his BSc degree from the Technical University of Catalonia, Spain, in 2011 and his MSc degree from Télécom ParisTech, France, in 2013. His research interests primary lie in computer vision, machine learning and its applications to sequential problems such as activity recognition from video data.

Dr. Tae-Kyun Kim Is an Assistant Professor and leader of Computer Vision and Learning Lab at Imperial College London, UK, since November 2010. He obtained his PhD from University of Cambridge in 2008 and Junior Research Fellowship (governing body) of Sidney Sussex College, University of Cambridge for 2007–2010. His research interests primarily lie in decision forests (tree-structure classifiers) and linear methods for articulated hand pose estimation, face analysis and recognition by image sets and videos, 6D object pose estimation, active robot vision, activity recognition, object detection/tracking, etc. which lead to novel active and interactive vision. He has co-authored over 40 academic papers in top-tier conferences and journals in the field; his co-authored algorithm for face image retrieval is an international standard of MPEG-7 ISO/IEC. He is co-recipient of the KUKA best service robotics paper award at ICRA14 and general co-chair of CVPR15 workshop on HANDS and ICCV15 workshop on Object Pose.