Recognizing Human Actions by using Effective Codebooks and Tracking

Lamberto Ballan, Lorenzo Seidenari, Giuseppe Serra, Marco Bertini and Alberto Del Bimbo

Abstract Recognition and classification of human actions for annotation of unconstrained video sequences has proven to be challenging because of the variations in the environment, appearance of actors, modalities in which the same action is performed by different persons, speed and duration and points of view from which the event is observed. This variability reflects in the difficulty of defining effective descriptors and deriving appropriate and effective codebooks for action categorization. In this chapter we present a novel and effective solution to classify human actions in unconstrained videos. In the formation of the codebook we employ radius-based clustering with soft assignment in order to create a rich vocabulary that may account for the high variability of human actions. We show that our solution scores very good performance with no need of parameter tuning. We also show that a strong reduction of computation time can be obtained by applying codebook size reduction with Deep Belief Networks with little loss of accuracy.

1 Introduction

With the continuous growth of video production and archiving, the need for automatic annotation tools that enable effective retrieval by content has accordingly gained increasing importance. In particular, action recognition is a very active research topic with many important applications such as human-computer interaction, video indexing and video-surveillance. Existing approaches for human action recognition can be classified as using holistic or part-based information [48, 3]. Most of the holistic-based methods usually perform better in a controlled environment and are also computationally expensive due to the requirement of pre-processing the input data. Moreover, these representations can be influenced by motions of multiple

The authors are with Media Integration and Communication Center, University of Florence, Italy (e-mail: lamberto.ballan@unifi.it; lorenzo.seidenari@unifi.it; serra@dsi.unifi.it; marco.bertini@unifi.it; alberto.delbimbo@unifi.it); Viale Morgagni 65, 50134 - Firenze, Italy.

objects, variations in the background and occlusions. Instead, part-based representations that exploit interest point detectors combined with robust feature descriptors, have been used very successfully for object and scene classification tasks in images [15, 62]. As a result, nowadays most video annotation solutions have exploited the bag-of-features approach to generate textual labels that represent the categories of the main and easiest to detect entities (such as objects and persons) in the video sequence [50, 19].

The definition of effective descriptors that are able to capture both spatial and temporal features has opened the possibility of recognizing dynamic concepts in video sequences. In particular, interesting results have been obtained in the definition of solutions to automatically recognize human body movements, which usually represent a relevant part of video content [40, 43, 42, 52]. However, the recognition and classification of such dynamic concepts for annotation of generic video sequences has proven to be very challenging because of the very many variations in environment, people and occurrences that may be observed. These can be caused by cluttered or moving background, camera motion and illumination changes; people may have different size, shape and posture appearance; semantically equivalent actions can manifest differently or partially, due to speed, duration or self-occlusions; the same action can be performed in different modes by different persons. This great variability on the one hand reflects in the difficulty of defining effective descriptors and on the other makes it hard to obtain a visual representation that may describe such dynamic concepts appropriately and efficiently. Furthermore, these part-based approaches usually do not attempt to localize and track actions that is necessary in video surveillance applications.

1.1 Effective Spatio-Temporal Descriptors

Holistic descriptors of body movements have been proposed by a few authors. Among the most notable solutions, Bobick *et al.* [6] proposed motion history images and their low-order moments to encode short spans of motion. For each frame of the input video, the motion history image is a gray scale image that records the location of motion; recent motion results into high intensity values whereas older motion produces lower intensities. Efros *et al.* [14] created stabilized spatio-temporal volumes for each action video segment and extracted a smoothed dense optic flow field for each volume. They have proved that this representation is particularly suited for distant objects, where the detailed information of the appearance is not available. Yilmaz and Shah [60] used a spatio-temporal volume, built stacking object regions; descriptors encoding direction, speed and local shape of the resulting 3D surface were generated by measuring local differential geometrical properties. Gorelick *et al.* [18] analyzed three-dimensional shapes induced by the silhouettes and exploited the solution to the Poisson equation to extract features, such as shape structure and orientation. Global descriptors that jointly encode shape and motion were suggested in Lin *et al.* [30]; Wang *et al.* [55] exploited global histograms of optic flow together

with hidden conditional random fields. Although encoding much of the visual information, these solutions have shown to be highly sensitive to occlusions, noise and change in viewpoint. Most of them have also proven to be computationally expensive due to the fact that some pre-processing of the input data is needed, such as background subtraction, segmentation and object tracking. All these aspects make these solutions only suited for representation of body movements in videos taken in controlled contexts.

Local descriptors have shown better performance and are in principle better suited for videos taken in both constrained and unconstrained contexts. They are less sensitive to partial occlusions and clutter and overcome some of the limitations of the holistic models, such as the need of background subtraction and target tracking. In this approach, local patches at spatio-temporal interest points are used to extract robust descriptors of local moving parts and the bag-of-features approach is employed to have distinctive representations of body movements. Laptev [27] and Dollár [13] approaches have been among the first solutions. Laptev [27, 45] proposed an extension to the Harris-Förstner corner detector for the spatio-temporal case; interesting parts were extracted from voxels surrounding local maxima of spatio-temporal corners, i.e. locations of videos that exhibit strong variations of intensity both in spatial and temporal directions. The extension of the scale-space theory to the temporal dimension permitted to define a method for automatic scale-selection. Dollár *et al.* [13] proposed a different descriptor than Laptev's, by looking for locally periodic motion. While this method produces a denser sampling of the spatio-temporal volume, it does not provide automatic scale-selection. Despite of it, experimental results have shown that it improves with respect to [45].

Following these works, other authors have extended the definition of local interest point detectors and descriptors to incorporate time or combined static local features with other descriptors so to model the temporal evolution of local patches. Sun *et al.* [51] have fused spatio-temporal SIFT points with holistic features based on Zernike moments. In [56], Willems *et al.* extended SURF feature to time and defined a new scale-invariant spatio-temporal detector and descriptor that showed high efficiency. Scovanner *et al.* [46], have proposed to use grouping of 3D SIFT, based on co-occurrence, to represent actions. Kläser *et al.* [24] have proposed a descriptor based on histograms of oriented 3D gradients, quantized using platonic solids. Gao *et al.* [16] presented MoSIFT, an approach that extend the SIFT algorithm to find visually distinctive elements in the spatial domain. It detects spatio-temporal points with a high amount of optical flow around the distinctive points motion constraints. More recently, Laptev *et al.* [28] proposed a structural representation based on dense temporal and spatial scale sampling, inspired by the spatial pyramid approach of [29] with interesting classification results in generic video scenes. Kovashka *et al.* [26] extended this work by defining a hierarchy of discriminative neighborhoods instead of using spatio-temporal pyramids. Liu *et al.* [32] combined MSER and Harris-Affine [38] regions with Dollár's space-time features and used AdaBoost to classify YouTube videos. Shao *et al.* [47] applied transformation based techniques (i.e. Discrete Fourier Transform, Discrete Cosine Transform and Discrete Wavelet Transform) on the local patches and used the transformed coefficients

as descriptors. Yu *et al.* [61] presented good results using the Dollar's descriptor and random forest-based template matching. Niebles *et al.* [41] trained an unsupervised probabilistic topic model using the same spatio-temporal features, while Cao *et al.* [8] suggested to perform model adaptation in order to reduce the amount of labeled data needed to detect actions in videos of uncontrolled scenes. Comparative evaluations of the performance of the most notable approaches were recently reported by Wang *et al.* [54] and Shao *et al.* [48].

1.2 Suitable Visual Codebooks

According to the bag-of-features model actions are defined as sets of codewords obtained from the clustering of local spatio-temporal descriptors. Most of the methods have used the k-means algorithm for clustering because of its simplicity and speed of convergence [41, 15, 49, 22]. However, both the intrinsic weakness of k-means to outliers and the need of some empirical pre-evaluation of the number of clusters hardly fit with the nature of the problem at hand. Moreover, with k-means the fact that cluster centers are selected almost exclusively around the most dense regions in the descriptor space results into ineffective codewords of action primitives. To overcome the limitations of the basic approach, Liu *et al.* [33] suggested a method to automatically find the optimal number of visual word clusters through maximization of mutual information (MMI) between words and actions. MMI clustering is used after k-means to discover a compact representation from the initial codebook of words. They showed some performance improvement. Recently Kong *et al.* [25] have proposed a framework that unifies reduction of descriptor dimensionality and codebook creation, to learn compact codebooks for action recognition optimizing class separability. Differently, Uemura and Mikolajczyk [39] explored the idea of using a large number of features represented in many vocabulary trees instead of a single flat vocabulary. Yao *et al.* [59] recently proposed a similar framework using a training procedure based on a Hough voting forest. Both these methods require higher efforts in the training phase.

1.3 Our Contribution

In this chapter we propose a novel and effective solution to classify human actions in unconstrained videos. It improves on previous contributions in the literature through the definition of a novel local descriptor and the adoption of a more effective solution for the codebook formation. We use image gradient and optic flow to respectively model the appearance and motion of human actions at regions in the neighborhood of local interest points and consider multiple spatial and temporal scales. These two descriptors are used in combination to model local features of human actions and activities. Unlike similar related works [46, 24], no parameter tuning is required.

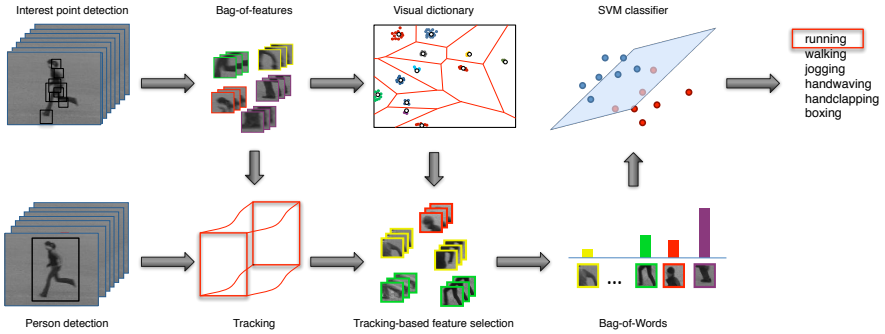


Fig. 1 The proposed solution architecture.

In the formation of the codebook we recognize that the clusters of spatio-temporal descriptors should be both in a sufficiently large number and sufficiently distinguished from each other so to represent the augmented variability of dynamic content with respect to the static case. To this end radius-based clustering [23] with soft assignment has been used. In fact, with radius-based clustering cluster centers are allocated at the modes corresponding to the maximal density regions, so resulting into a statistics of the codewords that better fits with the variability of human actions with respect to k-means clustering. To obtain a precise spatio-temporal localization of each action, the detected spatio-temporal points are associated to each person, present in the scene, by a particle filter visual tracker. Experiments carried on standard datasets show that the approach followed outperforms the current state of the art methods. To avoid too large codebooks we performed codebook compression with Deep Belief Networks. The solution proposed shows good accuracy even with very small codebooks. Finally, we provide several experiments on the Hollywood2 dataset [36] and on a new surveillance dataset (MICC-Surveillance), to demonstrate the effectiveness and generality of our method for action recognition in unconstrained video domains.

The rest of the chapter is organized as follows[1]. The full framework of the proposed solution is shown in Section 2, while the spatio-temporal features are presented in Section 3. Action representation and categorization is presented in Section 4. The experimental results, with an extensive comparison with the state-of-the-art approaches, are hence discussed in Section 5. Here we also included experiments on unconstrained videos to demonstrate the effectiveness of the approach also in this case. Conclusions are drawn in Section 6.

[1] Please note that an earlier version of this work has recently appeared in IEEE Transactions on Multimedia [4].

2 Action Classification Architecture

The architectural design of the proposed solution, based on an effective bag-of-features model, is shown in Fig. 1.

2.1 Visual Dictionary Formation

The basic idea of the bag-of-features model is to represent visual content as an unordered collection of "visual words". To this end, it is necessary to define a visual dictionary from the local features extracted in the video sequences, performing a quantization of the original feature space. The descriptors used to represent the spatio-temporal interest points are presented in the following Section 3.

The visual dictionary (codebook) is generated by clustering of a set of local descriptors and each cluster is treated as a visual word. Typically it is used the k-means algorithm because of its simplicity and convergence speed. However, it has been shown that using this algorithm the cluster centers tend to coalesce around the denser regions of the feature space, thus not describing other informative regions. This issue is particularly important in the densely sampled space of the spatio-temporal features used in our approach. In the work of Jurie and Triggs [23] it has been shown that a different approach, namely radius-based clustering, is able to generate better visual dictionaries for the images that arise in natural scenes. We have therefore used a radius-based clustering technique, following a mean-shift approach [11], that improves the performance of the system over k-means. This issue is presented in detail in Section 4.

2.2 Person Tracking and Data Association

Person tracking is used to assign the detected spatio-temporal interest points to each person present in a video, to localize both in space and time each recognized action. The tracker adopted in our system implements a particle filter based tracking algorithm, presented by [2], that tracks position, size and speed of the target, describing the target appearance with its color histogram (using hue and saturation channels). The tracker is initiated using the human detector of [12], implemented in OpenCV. The detector is run frame-wise to obtain both new targets to follow and measures for existing tracks. Measures obtained from the people detector are associated to targets by solving a data association problem, using a fast greedy algorithm that has a much lower complexity than the optimal solution obtainable with the Hungarian algorithm [58]. This greedy algorithm can be executed in real-time, as needed in video-surveillance applications, and works as follows: a matrix M that contains all the matching scores $m_{i,j}$ between the i_{th} target and the j_{th} measure of the person detector is computed. The matching score is computed as:

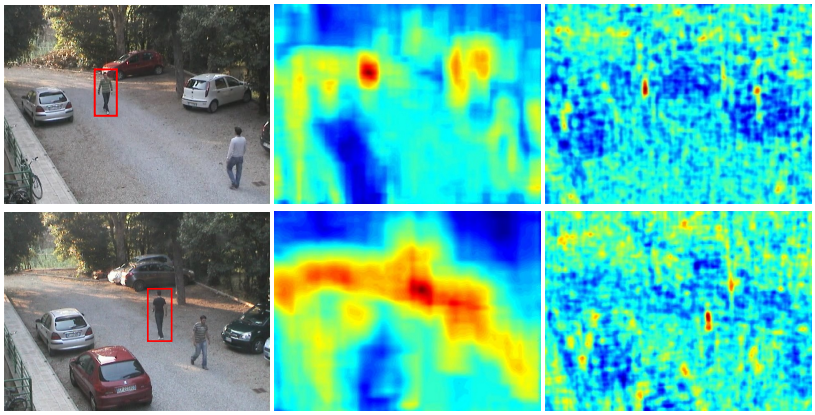


Fig. 2 Original frame, hue/saturation histogram and person detector generated likelihood computed for the farthest target (highlighted with red bounding box). In this example the pedestrian detector is run at a single scale; histogram likelihood is generated using the values of the Batthacharya distance between the template histogram and a corresponding (same scale and aspect ratio) window. In both cases scale and aspect ratio variations are not considered, for the sake of visualization.

$$m_{i,j} = e^{-\frac{d_{i,j}^2}{D}} \tag{1}$$

where $d_{i,j}$ is the Euclidean distance between the static part of the model (position and size) of the target and the position and size of the detected person (represented using top-left and bottom-right coordinates of the bounding boxes) and D is adaptively chosen based on the target size.

The maximum $m_{i,j}$ are iteratively selected, and the i rows and j columns belonging to target and detector in M are deleted. This is repeated until no further valid $m_{i,j}$ is available. To avoid the erroneous association of a detection to a target two approaches are followed: *i)* only the associated detections with a matching score $m_{i,j}$ above a threshold are used, to avoid that a detection that is far from a target is matched; *ii)* if a detection overlaps more than one target no association is performed. If a detection is not associated to any target and does not overlap any existing target then it is used to start a new track.

The template of the target appearance is updated every time a new detection is associated to the track. In this way we prevent template drift and we allow the color histogram to adapt with respect to illumination changes and maneuvers which can change the target appearance. The state update equation, defined over the 8-dimensional state vector x_k (composed by 4 components for position and size and 4 components for their velocities), realizes a 1^{st}-order dynamic model:

$$x_k = Ax_{k-1} + v_{k-1}, A = \begin{bmatrix} I_4 & I_4\Delta t \\ 0 & I_4 \end{bmatrix}, \tag{2}$$

Fig. 3 Example of multiple person tracking, spatio-temporal interest point detection and their association to the tracks.

where I_4 is an 4×4 identity matrix, Δt is the time step and v_{k-1} is an additive, zero mean, isotropic Gaussian uncertainty term that represents the uncertainty in the state update. This uncertainty is parametrized in terms of the standard deviation on each component of the state vector. The measurement model exploits the results of the person detector whenever they are available.

The person detector likelihood is strongly peaked in presence of a target, as shown in the third column of Fig. 2. This behavior allows to detect as distinct objects even very close pedestrians, but is not suitable to use it as likelihood of the target [1] since in particle weight computation it could assign very high weights to a few or no particles, and almost uniform low weights to the remaining population, leading thus to a degeneracy problem. To deal with this issue the target model of the particle filter is based on the color histogram of the tracked object, aiming at robustness against non-rigidity, rotation and partial occlusion; after updating the template histogram with the new measure histogram, weights are computed according to the Batthacharya distance between the particle and the template histograms. On the other hand the color histogram is too weak to be used as an aspect model in a real-world video-surveillance scenario and should not be used as a sole measurement provider, as shown in the second column of Fig. 2; this is due to background pixels contaminating the template and the lack of discriminativity of the histogram caused also by its subsampling (we used eight hue bins and eight saturation bins, to reduce sensitivity to light conditions).

To improve the particle filter capability to effectively track the target, even if its appearance is not strongly characterized, the tracking method implements a particular technique, based on the use of the similarity of the current estimate with the original target histogram as an index of tracking quality, to manage the uncertainty in the state update equation by means of on-line adaptation of the error v_{k-1}. In particular, let us consider the case where the variances of position and size of the target are set to very high values. In this case the filter samples over a wide enough area to maximize the possibility of capturing the target in case of erratic changes in direction or velocity. The pitfall in this strategy, however, is that it also increases the likelihood that the particle filter will become distracted by spurious similar patches

in the background. Considering also the variances of the velocities the problem is even worse: from equation 2, in the update rule for propagating a particle from time $k-1$ to k, the uncertainty in the dynamic component is propagated to the static component. To reduce this effect a *blindness* value is computed by passing the similarity of estimate and original target histogram through a sigmoid; this *blindness* value is used to adjust the variances in such a way that the noise in the static component of the state observations is never amplified by the noise in the dynamic components. This allows the tracker to switch between two different behaviors: one that relies on the predicted motion of the target and one that behaves like a random-walk model.

2.3 Action Classification and Track Annotation

By mapping the features associated to each tracked person in a video to the vocabulary, we can represent it by the frequency histogram of visual words. In order to reduce outliers, histograms of tracks that contain too few interest points, are discarded. Then, the remaining histograms are fed to a classifier to predict the action category. In particular, classification is performed using non-linear SVMs with the χ^2 kernel. To perform multi-class classification we use the *one-vs-one* approach. To this end we train a binary SVM classifier for each pair of action classes for a total of $\frac{n(n-1)}{2}$ classifiers. Action is predicted considering the output of each SVM as a vote for the correspondent action and using a majority voting procedure. Fig. 3 shows an example of the tracker results and features association.

3 Fusing Spatio-temporal Local Descriptors of Appearance and Motion

Spatio-temporal interest points are detected at video local maxima of the Dollár's detector [13] applied over a set of spatial and temporal scales. Using multiple scales is fundamental to capture the essence of human activity. To this end, linear filters are separately applied to the spatial and temporal dimension: on the one hand, the spatial scale permits to detect visual features of high and low detail; on the other, the temporal scale allows to detect *action primitives* at different temporal resolutions. The filter response function is defined as:

$$R = \left(I * g_\sigma * h_{ev}\right)^2 + \left(I * g_\sigma * h_{od}\right)^2 \tag{3}$$

where $I(x,y,t)$ is the image sequence, $g_\sigma(x,y)$ is a spatial Gaussian filter with scale σ, h_{ev} and h_{od} are a quadrature pair of 1D Gabor filters that provide a strong response to temporal intensity changes for periodic motion patterns, respectively defined as:

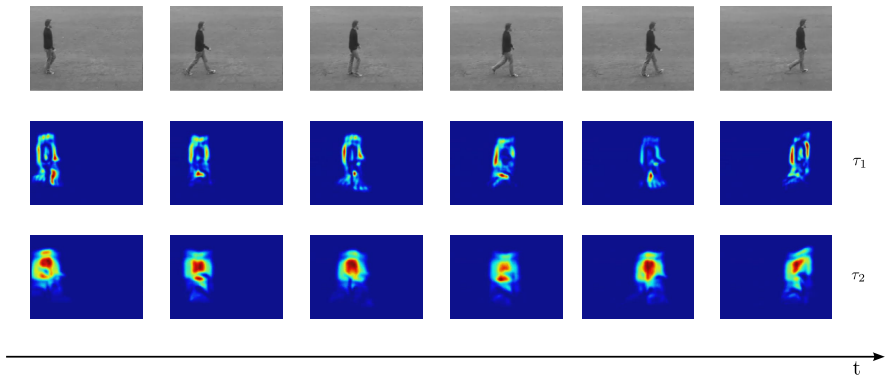


Fig. 4 Response of the spatio-temporal interest point detector at two temporal scales $\tau_1 < \tau_2$ (low response in blue, high response in red); first row: original video frames, second row detector response at temporal scale τ_1 (mostly due to motion of human limbs); third row: detector response temporal scale τ_2 (mostly due to motion of human torso).

$$h_{ev}(t;\tau,\omega) = -\cos(2\pi t\omega)e^{-t^2/\tau^2} \tag{4}$$

$$h_{od}(t;\tau,\omega) = -\sin(2\pi t\omega)e^{-t^2/\tau^2} \tag{5}$$

where $\omega = 4/\tau$. In the experiments we used $\sigma = \{2,4\}$ as spatial scales and $\tau = \{2,4\}$ as temporal scales. Fig. 4 shows an example of temporal scaling of human body parts activity during walking: torso has high response at high temporal scale, while limbs respond at the lower scale.

Three-dimensional regions of size proportional to the detector scale (6x) are considered at each spatio-temporal interest point, and divided into equally sized sub-regions (three for each spatial dimensions along the x and y, and two for the temporal dimension t), as shown in Fig. 5.

For each sub-region, image gradients on x, y and t are computed as:

$$G_x = I(x+1,y,t) - I(x-1,y,t) \tag{6}$$

$$G_y = I(x,y+1,t) - I(x,y-1,t) \tag{7}$$

$$G_t = I(x,y,t+1) - I(x,y,t-1) \tag{8}$$

and the optic flow with relative apparent velocity V_x, V_y is estimated according to [34].

Orientations of gradients and optical flow are computed for each pixel as:

$$\phi = \tan^{-1}\left(G_t/\sqrt{G_x^2+G_y^2}\right) \in \left[-\frac{\pi}{2},\frac{\pi}{2}\right] \tag{9}$$

$$\theta = \tan^{-1}\left(G_y/G_x\right) \in [-\pi,\pi] \tag{10}$$

$$\psi = \tan^{-1}\left(V_y/V_x\right) \in [-\pi,\pi] \tag{11}$$

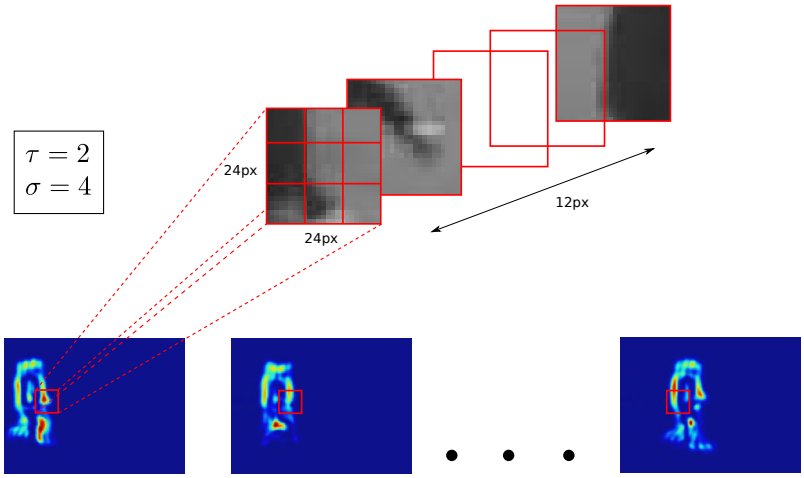


Fig. 5 Three dimensional region at the spatio-temporal interest point corresponding to a swinging arm.

where ϕ and the θ are quantized in four and eight bins, respectively.

The local descriptor obtained by concatenating ϕ and θ histograms (H3DGrad) has therefore size $3 \times 3 \times 2 \times (8+4) = 216$. There is no need to re-orient the 3D neighborhood, since rotational invariance, typically required in object detection and recognition, is not desirable in the action classification context. This approach is much simpler to compute than those proposed in [46] and [24]. In particular, in [46] the histogram is normalized by the solid angle value to avoid distortions due to the polar coordinate representation (instead of quantizing separately the two orientations as in our approach), moreover the size of the descriptor is 2048; in [24] the 3D gradient vector is projected on the faces of a platonic solid. In this latter approach requires additional parameter tuning, to optimize the selection of the solid used for the histogram computation and whether to consider the orientations of its faces or not. Differently from [28] our 12-bin H3DGrad descriptor models the dynamic appearance of the three-dimensional region used for its computation, instead of being a 4-bin 2D histogram cumulated over time. A comparison between our H3DGrad descriptor and the other HOG features (i.e. [24, 28, 46]) is reported in Table 1, in terms of both accuracy and feature computation time.

The ψ is quantized in eight bins with an extra "no-motion" bin added to improve performance. The local descriptor of ψ (HOF) has size $3 \times 3 \times 2 \times (8+1) = 162$. Histograms of ϕ, θ and ψ are respectively derived by weighting pixel contributions respectively with the gradient magnitude $M_G = \sqrt{G_x^2 + G_y^2 + G_t^2}$ (for ϕ and θ), and the optic flow magnitude $M_O = \sqrt{V_x^2 + V_y^2}$ (for ψ).

In order to obtain an effective codebook for human actions these two descriptors can be combined according to either early or late fusion. In the former case the two descriptors are first concatenated and the combined descriptor is hence used for the

Descriptor	KTH	Weizmann	Time (ms)
H3DGrad	90.38	92.30	1
Kläser *et al.* [24]	91.40	84.30	2
Laptev *et al.* [28]	81.60	-	12
Scovanner *et al.* [46]	-	82.60	419

Table 1 Comparison of accuracy and efficiency of our H3DGrad with other gradient based descriptors on KTH and Weizmann datasets. Computation time for a single descriptor measured on a 2.66GHz Intel Xeon with 12 GB RAM; H3DGrad, [24] and [28] are C++ implementations while [46] is a MATLAB implementation.

definition of the human action codebook. In the latter a codebook is obtained from each descriptor separately; then the histograms of codewords are concatenated to form the representation (see Fig. 6).

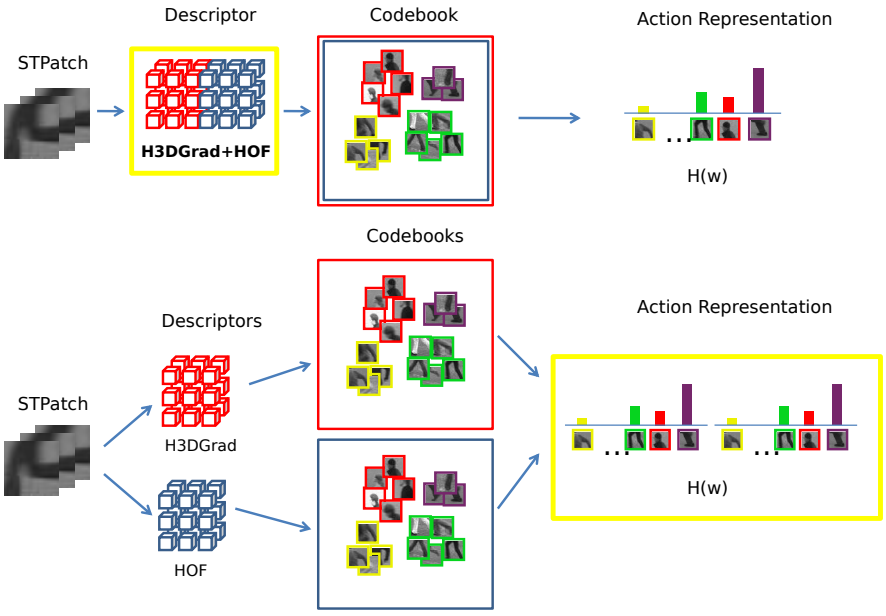


Fig. 6 Two fusion strategies: early-fusion (at the descriptor level) and late-fusion (at the codebook level).

Fig. 7 shows the classification accuracy measured with the KTH dataset, using codebooks based on the H3DGrad descriptor (a), HOF descriptor (b), and early (c) and late fusion (d), with 4000 codewords. Each action, is represented by an histogram H of codewords w obtained according to k-means clustering with hard assignment:

$$H(w) = \frac{1}{n}\sum_{i=1}^{n}\begin{cases}1 & \text{if } w = \underset{v\in V}{\operatorname{argmin}}(D(v,f_i));\\ 0 & \text{otherwise;}\end{cases} \tag{12}$$

where n is the number of the spatio-temporal features, f_i is the i-th spatio-temporal feature, and $D(v, f_i)$ is the Euclidean distance between the codeword v of the vocabulary V and f_i.

We present in Table 2 the average accuracy obtained by H3DGrad and HOF respectively, and by the early and late fusion. From the figures, it appears clearly that late fusion provides the best performance. This can be explained with the fact that H3DGrad and HOF descriptors have quite complementary roles (for example the *boxing* action is better recognized when using H3DGrad descriptor while *hand-clapping* action is better recognized by HOF, as shown in Fig. 7 (a),(b)). Late fusion improves recognition performance for all the classes except one. A similar behavior was observed with the Weizmann dataset, although in this case the improvement was not so significant mainly due to the limited size and intra-class variability of the dataset (see Table 2).

Descriptor	**KTH**	**Weizmann**
H3DGrad	90.38	92.30
HOF	88.04	89.74
H3DGrad + HOF (early fusion)	91.09	92.38
H3DGrad + HOF (late fusion)	**92.10**	**92.41**

Table 2 Average class accuracy of our descriptors, alone and combined, on the KTH and Weizmann datasets.

4 Action Representation and Classification

In order to improve with respect to k-means and to account for the high variability of human actions in terms of appearance or motion we used radius-based clustering for codebook formation.

Fig. 8 shows the codeword frequency of radius-based clustering and k-means with hard quantization on the KTH dataset. It is interesting to note that with k-means most of the codewords have similar probability of occurrence, so making it difficult to identify a set of words that have at the same time high discrimination capability and good probability of occurrence. In contrast radius-based shows a much less uniform frequency distribution. Interestingly, with radius-based clustering, the codeword distribution of the human action vocabulary is similar to the Zipf's law for textual corpuses. It seems therefore reasonable to assume that codewords at intermediate frequencies are the most informative also for human action classification, and the best candidates for the formation of the codebook.

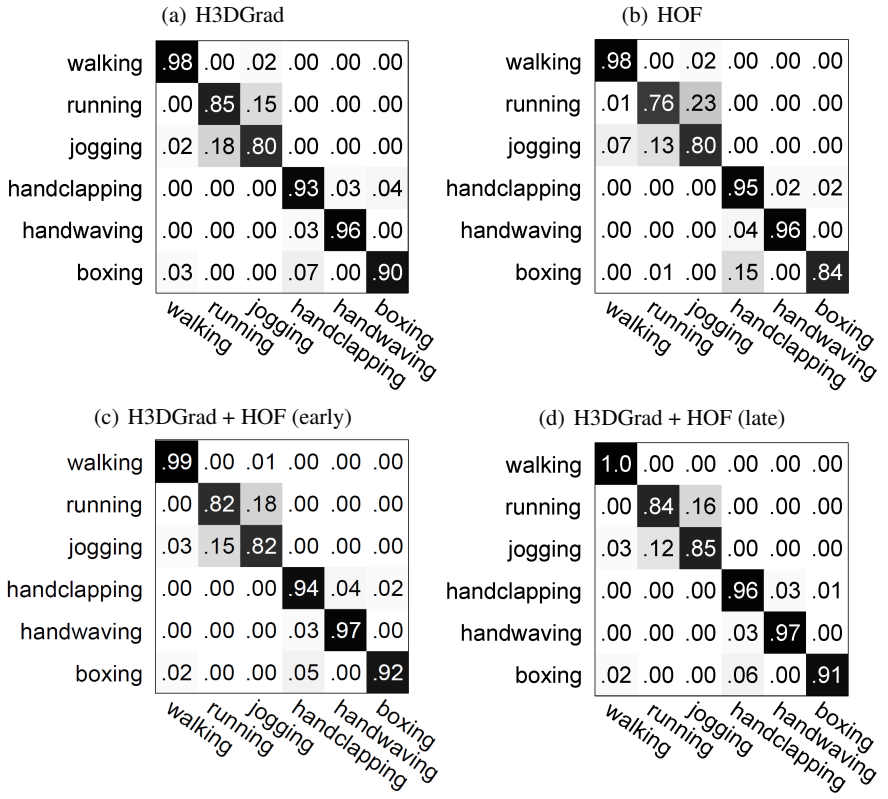


Fig. 7 Classification accuracy on the KTH dataset using k-means clustering, hard assignment and different descriptors combination strategies (i.e. early or late fusion).

Due to the high dimensionality of the descriptor, codebooks for human actions usually have cluster centers that are spread in the feature space, so that two or more codewords are equally relevant for a feature point (codeword *uncertainty*); moreover cluster centers are often too far from feature points so that they are not anymore representative (codeword *plausibility*). With radius-based clustering, codeword *uncertainty* is critical because it frequently happens that feature points are close to the codewords boundaries [17]. Instead, codeword *plausibility* is naturally relaxed due to the fact that clusters are more uniformly distributed in the feature space. To reduce the *uncertainty* in codeword assignment, we therefore performed radius-based clustering with soft assignment by Gaussian kernel density estimation smoothing. In this case, the histogram H is computed as:

$$H(w) = \frac{1}{n} \sum_{i=1}^{n} \frac{K_\sigma(w, f_i)}{\sum_{j=1}^{|V|} K_\sigma(v_j, f_i)} \tag{13}$$

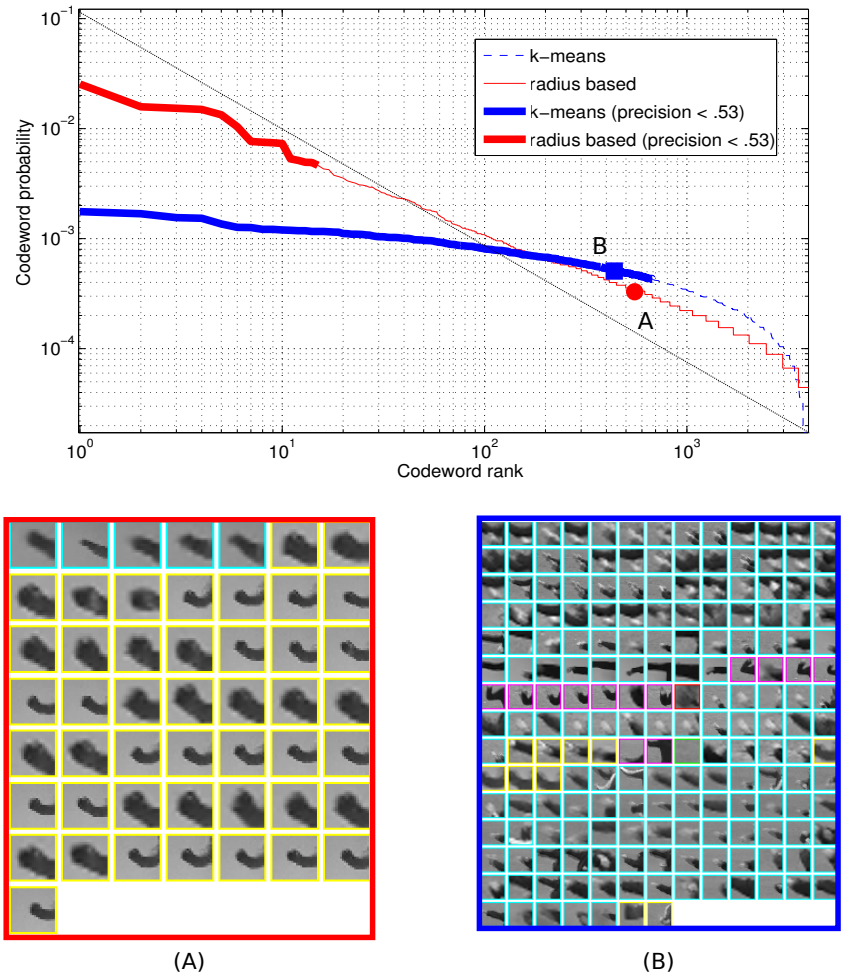


Fig. 8 Log-log plots of codeword frequency using k-means and radius-based clustering with hard assignment. Bold lines indicate regions where the average cluster precision [37] is below 0.53. The dotted diagonal line represents the Zipfian distribution. Two sample clusters are shown at near frequencies, respectively obtained with radius-based clustering (A) (most of the features in the cluster represent spatio-temporal patches of the same action) and with k-means (B) (features in the cluster represent patches of several actions). Patches of actions have different colors: *boxing* (cyan), *hand-waving* (magenta), *hand-clapping* (yellow), *running* (green), *walking* (red), *jogging* (blue).

where K_σ is the Gaussian kernel: $K_\sigma(\cdot,\cdot) = \frac{1}{\sqrt{2\pi}\sigma} e^{\left(-\frac{d(\cdot,\cdot)^2}{2\sigma^2}\right)}$ being σ the scale parameter tuned on the training set, and $d(\cdot,\cdot)$ is the Euclidean distance.

Fig. 9 compares the classification accuracy with codebooks obtained with k-means clustering with both hard and soft assignment, and radius-based clustering

with soft assignment, respectively for the KTH and Weizmann dataset. The plots have been obtained by progressively adding less frequent codewords to the codebooks (respectively up to 4000 and 1000 codewords for the two datasets). The performance of k-means is improved by the use of soft assignment. With a small number of words radius-based clustering with soft assignment has lower performance than k-means due to the fact that the codewords used have higher frequency than those used by k-means (see Fig. 8). As the number of codewords in the codebook increases, radius-based clustering outperforms k-means, whether with hard or soft assignment. This reflects the fact that in this case radius-based clustering permits to have also sparse regions being represented in the codebook. Besides, soft assignment helps to reduce *uncertainty* in the dense regions. Fig. 10 shows the confusion matrix for different human actions on KTH and Weizmann datasets with radius-based soft assignment. The average accuracy is respectively 92.66% and 95.41% for the two datasets.

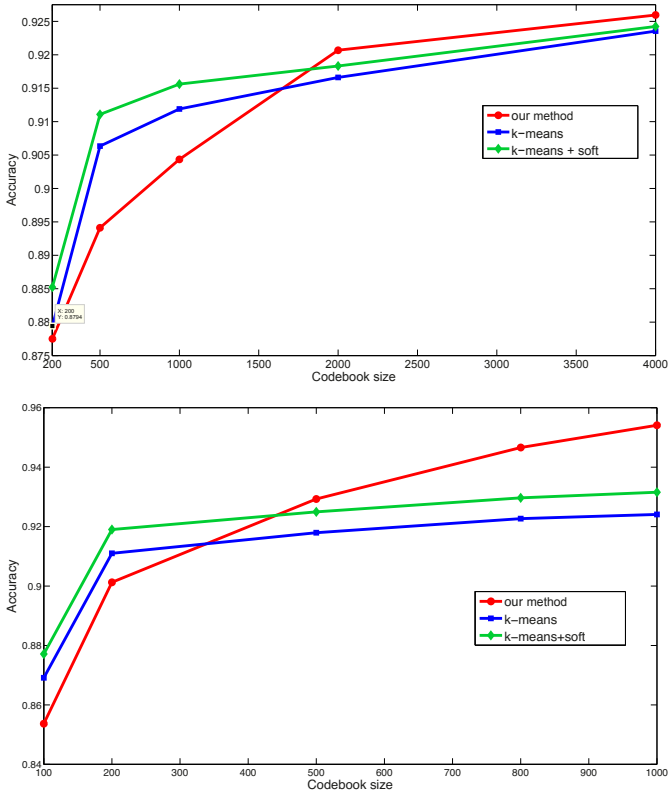


Fig. 9 Classification accuracy on KTH (top) and Weizmann (bottom) datasets with codebooks created with k-means with hard assignment, k-means with soft assignment and radius-based with soft assignment.

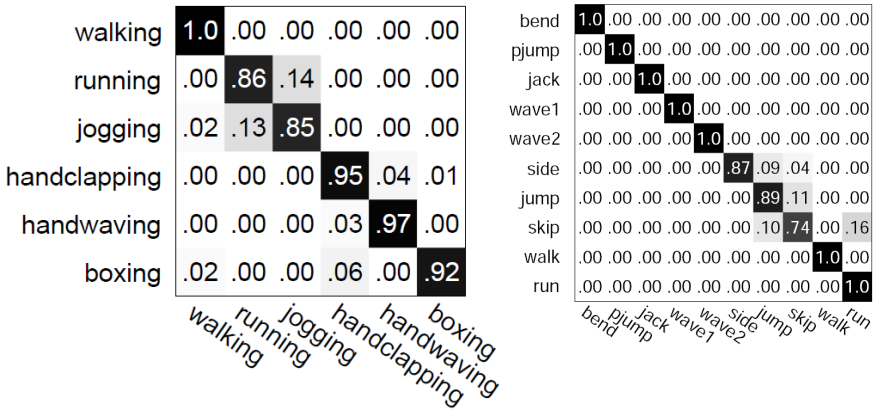


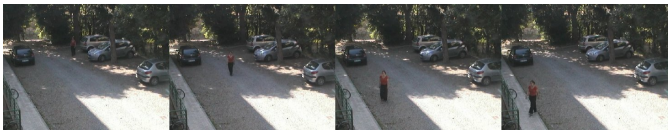
Fig. 10 Classification accuracy on KTH (left) and Weizmann (right) datasets using radius-based clustering with soft assignment.

5 Experimental Results

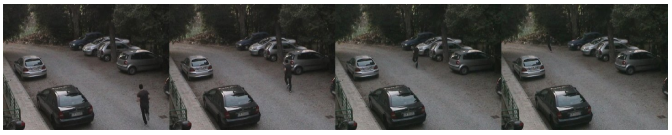
We have assessed our approach for categorization of human actions in different conditions. Particularly, it has been tested on the KTH and Weizmann datasets that show staged actions performed by an individual in a constrained non-cluttered environment. Moreover, in order to have a more complete assessment of the performance of the proposed solution even in real world scenes with high variability and unconstrained videos, we also carried out experiments on the Hollywood2 and MICC-UNIFI Surveillance datasets. This latter, made publicly available at *www.openvisor.org* [53], includes real world video surveillance sequences containing actions performed by individuals with cluttering and varying filming conditions. Experiments were performed using non-linear SVMs with the χ^2 kernel [62].

5.1 Experiments on KTH and Weizmann datasets

The KTH dataset, currently the most common dataset used for the evaluations of action recognition methods [54], contains 2391 short video sequences showing six basic actions: *walking, running, jogging, hand-clapping, hand-waving, boxing*. They are performed by 25 actors under four different scenarios with illumination, appearance and scale changes. They have been filmed with a hand-held camera at 160×120 pixel resolution. The Weizmann dataset contains 93 short video sequences showing nine different persons, each performing ten actions: *run, walk, skip, jumping-jack, jump-forward-on-two-legs, jump-in-place-on-two-legs, gallop-sideways, wave-two-hands, wave-one-hand* and *bend*. They have been filmed with a fixed camera, at 180×144 pixel resolution, under the same lighting condition.

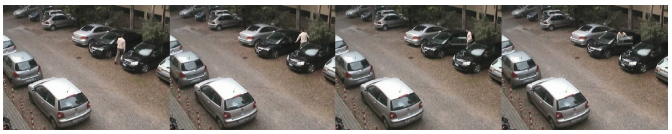


(a) Walking



(b) Running

(c) Pickup object



(d) Enter car

(e) Enter car (from a different view point)

(f) Exit car

(g) Handshake

(h) Give object

Fig. 11 Sample frames of sequences from the MICC-UNIFI Surveillance dataset.

Table 3 reports the average accuracy of our method in comparison with the most notable research results published in the literature. The performance figures reported are those published in their respective papers. For a fair comparison, our experiments have been performed with the setup suggested by the creators of the KTH and Weizmann datasets [45, 18], that has been used in [61, 31, 51, 16, 44, 28, 57, 46, 24, 56, 45, 54]. In particular, with the KTH dataset, SVM classifiers have been trained on sequences of 16 actors and performance was evaluated for the sequences of the remaining 9 actors according to 5-fold cross-validation. With the Weizmann dataset SVM classifiers have been trained on the videos of 8 actors and tested on the one remaining, following leave-one-out cross-validation.

While showing the best performance, our solution has also the nice property that it does not require any adaptation to the context under observation. Instead other solutions require some tuning of the descriptor to the specific context. Namely, Laptev *et al.* [28] perform different spatio-temporal sampling of video frames and define a set of descriptors; hence they represent each action with the best combination of sampling and descriptors; Kläser *et al.* [24] use a parameterized 3D gradient descriptor; parameter values are optimized for the dataset used; Liu *et al.* [31] use both local and global descriptors and select the best combination of them according to an optimization procedure; Scovanner *et al.* [46] optimize the codebook by associating co-occurrent visual words.

Other researchers have claimed higher performance on the KTH dataset: 93.17% Bregonzio *et al.* [7]; 94.2% Liu and Shah [33]; 93.43% Lin *et al.* [30]; 95.83% Chen *et al.* [10]. However, these results were obtained with a Leave-One-Out Cross-Validation setting that uses more training data and therefore are not directly comparable. For the sake of fairness, they have not been included in Table 3. An exhaustive list of the different experimental setups and results has been recently published by Gao *et al.* [16].

5.2 Experiments on MICC-UNIFI Surveillance dataset

The MICC-UNIFI Surveillance dataset is composed by 175 real world video sequences of human actions with durations ranging from 3 to 20 seconds. The videos have been taken from wall mounted Sony SNC RZ30 cameras at 640×480 pixel resolution, in a parking lot. The scenes are captured from different viewpoints, at different degrees of zooming, with different shadowing and unpredictable occlusions, at different duration, speed and illumination conditions. Eight subjects perform seven everyday actions: *walking, running, pickup object, enter car, exit car, handshake* and *give object*. A few examples are shown in Fig. 11. We followed a repeated stratified random sub-sampling validation, using 80% of the videos of each class as training set. Experiments were performed using a 2000 codeword codebook. The confusion matrix of classification accuracy is reported in Fig. 12: the average accuracy is 86.28%. Most of the misclassifications observed with our method occurred with the *give object* and *handshake* actions. They are both characterized by

Method	KTH	Weizmann	Features	Optimizations
Our method	**92.66**	**95.41**	H3DGrad + HOF	-
Yu *et al.* [61]	91.8		HoG + HOF	-
Wang *et al.* [54]	92.1	-	HOF	-
Gao *et al.* [16]	91.14	-	MoSIFT	-
Sun *et al.* [51]	89.8	90.3	2D SIFT + 3D SIFT + Zernike	-
Rapantzikos *et al.* [44]	88.3	-	PCA-Gradient	-
Laptev *et al.* [28]	91.8	-	HoG + HOF	codebook, sampling
Dollár *et al.* [13]	81.2	-	PCA-Gradient	-
Wong and Cipolla [57]	86.62	-	PCA-Gradient	-
Scovanner *et al.* [46]	-	82.6	3D SIFT	codebook
Niebles *et al.* [41]	83.33	90	PCA-Gradient	-
Liu *et al.* [31]	-	90.4	PCA-Gradient + Spin images	codebook
Kläser *et al.* [24]	91.4	84.3	3D HoG	descriptor
Willems *et al.* [56]	84.26	-	3D SURF	-
Schüldt *et al.* [45]	71.7	-	ST-Jets	-

Table 3 Comparison of classification accuracy with some state-of-the-art methods on KTH and Weizmann datasets.

a very fast motion pattern and small motion of the human limbs. Fig. 13 reports sample sequences of these actions with evidence of details. In Table 4, we report a comparison of our method with other codebook creation approaches (k-means with hard and soft assignment) and with other state-of-the-art descriptors that publicly make their implementation available: MoSIFT[2] [16] and Dollár *et al.* [3] [13]. The results show that the proposed method outperforms the other approaches, and that the proposed codebook creation approach performs better than the typical k-means clustering whether with hard and soft assignment.

Method	MICC-Surveillance
Our method	**86.28**
k-means + soft	83.74
k-means	82.90
Dollár *et al.* [13]	72.50
MoSIFT [16]	75.88

Table 4 Comparison of classification accuracy on MICC-Surveillance dataset with our method, k-means with soft assignment, k-means with hard assignment, and with the descriptors proposed in [13] and [16].

[2] http://lastlaugh.inf.cs.cmu.edu/libscom/downloads.htm

[3] http://vision.ucsd.edu/%7epdollar/research.html

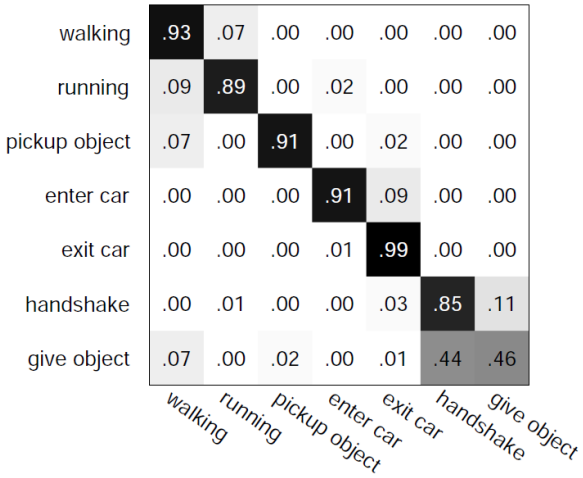


Fig. 12 Classification accuracy on the MICC-Surveillance dataset using radius-based clustering with soft assignment.

5.3 Experiments on Hollywood2 dataset

The Hollywood2 dataset [36] is composed by sequences extracted from DVDs of 69 Hollywood movies, showing 12 different actions in realistic and challenging settings: *answer phone, drive car, eat, fight person, get out of car, handshake, hug person, kiss, run, sit down, sit up, stand up*. We performed our experiments with the same setup of [28, 54] using the "clean" training dataset, containing scenes that have been manually verified. This dataset is composed by 1707 sequences divided in training set (823) and test set (884), with different frame size and frame rate; train and test set videos have been selected from different movies. To be comparable with other experimental results the performance has been evaluated computing the average precision (AP) for each class and reporting also the mean AP over all classes. Codebooks have been created using 4000 codewords, as in [54]. We have compared our codebook creation approach with k-means clustering using both soft and hard assignments, and with an implementation of the method proposed in [28] using the provided descriptor and detector[4]. Results are reported in Table 5, showing that the proposed method outperforms the other approaches in the majority of action classes and in terms of mean AP.

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

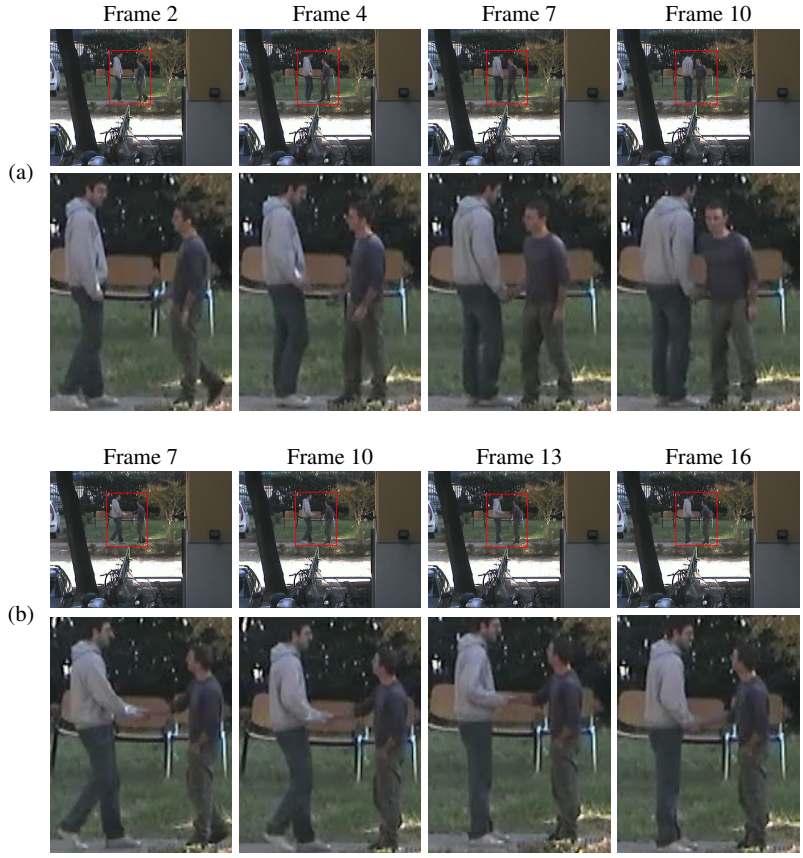


Fig. 13 Sample frames of *give object* (a) and *handshake* (b) action sequences in the MICC-Surveillance dataset. For each sequence the second row shows the detail indicated in red in the first row.

5.4 Experiments on Reducing the Codebook Size

Large codebooks, although being able to exploit the most informative codewords as illustrated in Fig. 8, imply high time and space complexity. Reduction of codebook size with preservation of descriptive capability is therefore desirable. Linear dimensionality reduction techniques such as Principal Component Analysis or Latent Semantic Analysis, are not suited to this end because they are not able to handle high order correlations between codewords that are present in human action representation [35]. We have therefore applied nonlinear dimensionality reduction with Deep Belief Networks (DBNs) [20, 35]. A DBN is composed of several Restricted Boltzmann Machines (RBM) building blocks that encode levels of non-linear relationships of the input vectors. It is pre-trained by learning layers incrementally using contrastive divergence [9]. After pre-training, the auto-encoder is built by reversing

Action	*k-means*	*k-means + soft*	*Our method*	Laptev *et al.* [28]
Answer phone	0.178	0.186	**0.195**	0.134
Drive car	0.864	**0.865**	0.863	0.861
Eat	0.552	0.564	0.564	**0.596**
Fight person	0.564	0.557	0.578	**0.643**
Get put of car	0.362	0.364	**0.362**	0.297
Handshake	0.142	0.143	0.167	**0.179**
Hug person	0.251	0.257	0.275	**0.345**
Kiss	0.494	**0.510**	0.503	0.467
Run	0.631	0.636	**0.659**	0.619
Sit down	0.483	0.493	**0.509**	0.505
Sit up	0.215	0.231	**0.227**	0.143
Stand up	0.511	0.513	**0.514**	0.485
mean AP	0.437	0.443	**0.451**	0.439

Table 5 Comparison of per-class AP performance on Hollywood2 dataset with codebooks created with our method, k-means with soft assignment, k-means with hard assignment and with the detector+descriptor proposed by Laptev *et al.* [28].

the network and connecting the top layer of the network to the bottom layer of its reversed version. The auto-encoder is then used to fine-tune the network using a standard back-propagation algorithm.

Since the action representation $H(w)$ can be considered as a coarse probability density estimation of the features of a human action (see equation 13), given a set of space-time features $\mathscr{F} = \{f_1, f_2, \ldots, f_n\}$, the value of the i-th bin of H can be considered as the probability that a space-time descriptor $f \in \mathscr{F}$ is represented by the codeword w_i. This probability can hence be used as an input for an RBM according to [21].

Fig. 14 reports plots of accuracy measured at different codebook sizes, with PCA, LSA and DBN codebook reduction and radius-based clustering with soft assignment, on the KTH dataset. Codebook reduction was applied to a 4000 codewords codebook. The dimension of the input layer is equal to the size of the uncompressed codebook and the dimension of the output layer is the compressed codebook size. Each hidden layer is one half the dimension of its input layer. The network depth ranges between five and eight depending on the size of the output codebook. The performance of our approach outperforms that of the method recently proposed in [25], especially for the smaller codebook sizes.

Fig. 15 reports plots of mean computation times for a KTH video sequence as a function of codebook size for radius-based clustering with soft assignment. The accuracy values of Fig. 14 have been reported on the plot for the sake of completeness. It can be noticed that strong codebook size reductions result into time improvements of more than two orders of magnitude. A compressed codebook with 100 codewords scores 89.57% recognition accuracy with respect to 92.66% of a 4000 codewords codebook.

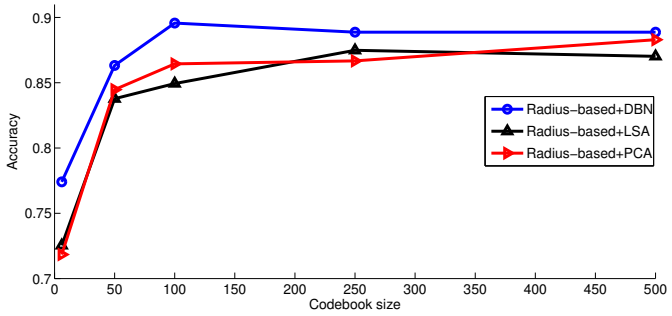


Fig. 14 Classification accuracy on KTH dataset at different codebook sizes, with different codebook reduction techniques, for radius-based clustering with soft assignment.

Fig. 16 shows that DBN-compressed codebooks on the one hand provide good accuracy even with very small codebook sizes, and on the other hand make radius-based clustering still competitive with respect to k-means clustering with 100 or less codewords.

Table 6 reports a comparison in terms of classification accuracy at different codebook sizes with DBN, PCA and LSA on the MICC-UNIFI surveillance dataset. Codebook reduction was applied to the 2000 codeword codebook obtained with radius-based clustering and soft assignment in the previous classification experiment. The smaller number of available training videos, with respect to KTH, is responsible for the reduction in classification accuracy, although the DBNs largely outperform the other methods. This experiment shows another advantage of the use of DBNs over PCA and LSA when the number of sequences available for training is relatively small, i.e. the possibility to create larger dictionaries that usually yield higher classification accuracy although maintaining a speed improvement of an order of magnitude. Table 7 reports a comparison of MAP performance obtained using compressed codebooks created with DBN, PCA and LSA on the Hollywood2 dataset. Codebook reduction was applied to the 4000 codeword codebook obtained with radius-based clustering and soft assignment used in the classification experiment. Despite the challenging dataset, the performance is still comparable with that obtained with full sized codebooks by several approaches reported in [54].

Codebook size	6	50	100	250	500
DBN	0.386	0.397	0.412	0.431	0.474
PCA	0.333	0.378	0.405	-	-
LSA	0.330	0.346	0.335	-	-

Table 6 Classification accuracy on MICC-UNIFI dataset at different codebook sizes, with different codebook reduction techniques, for radius-based clustering with soft assignment. Using PCA and LSA it is not possible to create codebooks larger than the number of training videos; using DBNs this issue is not present.

Codebook size	6	50	100	250	500
DBN	0.281	0.372	0.383	0.375	0.374
PCA	0.191	0.323	0.329	0.337	0.338
LSA	0.204	0.322	0.316	0.311	0.314

Table 7 Classification of MAP performance on Hollywood2 dataset at different codebook sizes, with different codebook reduction techniques, for radius-based clustering with soft assignment.

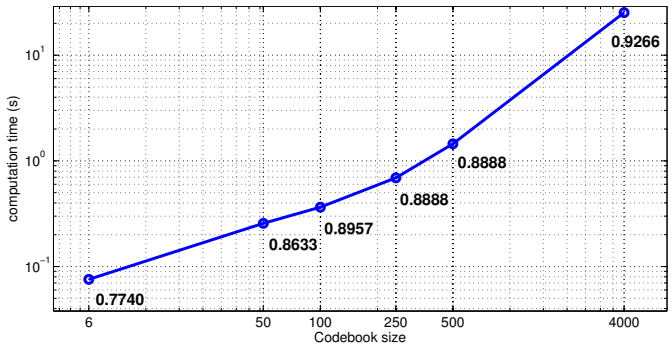


Fig. 15 Mean computation times for a KTH video sequence at different codebook sizes with radius-based clustering and DBNs. The numbers associated to the markers indicate the classification accuracy.

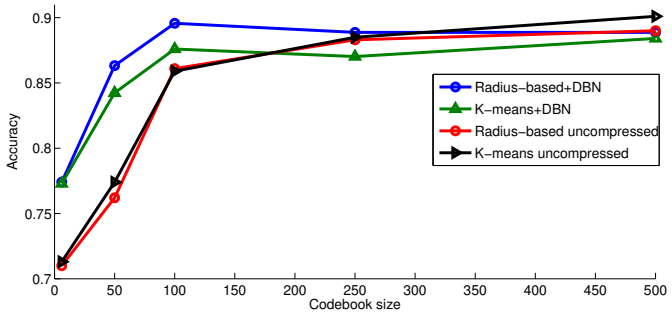


Fig. 16 Classification accuracy as a function of codebook size, for DBN-compressed and uncompressed codebooks. Radius-based clustering with soft assignment is compared with k-means clustering with hard assignment.

5.5 Tracker Evaluation and Experiments on Recognizing Multiple Actions

These experiments have been perfomed on a subset of the MICC-UNIFI Surveillance dataset. First of all, we have evaluated our tracking module quality by measuring multiple object tracking accuracy (MOTA) as defined by Bernardin and Stiefelhagen [5]. MOTA is an intuitive performance metric for multiple object trackers and measures a tracker performance at keeping accurate trajectories. For each frame processed a tracker should produce a set of object hypotheses, each of which should ideally correspond to a real visible object. In order to compute MOTA a consistent hypothesis-object mapping over time must be produced; the complete procedure to obtain this mapping is specified in detail in [5]. MOTA takes into account all possible errors that a multi-object tracker makes: false positives, missed objects and identity switches. False positives (fp) arise when, for example, the tracker is initiated on a false detection or when an object is missed and consequently a wrong pattern replaces the correct object hypothesis. Misses or false negatives (fn) arise whenever an object is not mapped to any of the hypotheses proposed by the tracker; finally identity switches (sw) happen whenever an object hypothesis is mapped to the wrong object, for example after an occlusion or when an object tracker fails and another tracker is reinitialized. Errors are normalized by the number of objects present (gt) with respect to the whole sequence.

MOTA is defined as follows:

$$MOTA = 1 - \frac{\sum_t fp_t + fn_t + sw_t}{\sum_t gt_t} \tag{14}$$

We represent persons as bounding boxes and we consider a mapping correct if $\frac{O \cap H}{O \cup H} \geq 0.5$, where O and H are the areas of the object and the hypothesis bounding boxes mapped. We measured MOTA for all five sequences in which our final recognition experiments were performed and another sequence. The last sequence is recorded with a PTZ camera, panning tilting and zooming on targets and targets are instructed to produce overlapping trajectories in order to create difficult situations for a multiple object tracker. In the first five test sequences most of the errors are caused by false alarms of the pedestrian detector that cause instantiation of trackers; in the classification stage this empty tracks can be filtered since they usually do not contain enough detected space-time interest points. In the last sequence most of the errors are due to identity switches since target maneuvers are more complex. MOTA is quite satisfying in all sequences, considering also that, in order to attain real-time performance, our appearance model is weak and no online classifier is used to perform data association or learn the template.

We have further evaluated the performance of our approach on five complex video sequences containing multiple actions performed concurrently (two examples are shown in Fig. 17). These sequences have different durations ranging from a minimum of ~ 120 to a maximum of ~ 300 frames. Our method has been applied to recognize and localize two basic actions: *walking* and *running*. As training set

Seq.	FPR	FNR	SWITCH	MOTA
1	27.92	2.92	0	68.35
2	38.56	12.40	2	49.82
3	13.15	32.16	0	54.67
4	23.65	9.18	0	67.20
5	15.02	27.48	0	57.74
6	14.59	3.82	52	79.38

Table 8 Multiple object tracking accuracy (MOTA) together with false positive rate (FPR), false negative rate (FPR) and amount of identity switches (SWITCH).

Seq.	**Detected**	**Filtered**	$\mathbf{W}_{\mathrm{GT}}$	$\mathbf{R}_{\mathrm{GT}}$	$\mathbf{O}_{\mathrm{GT}}$	**Acc**
1	8	5	3	2	0	4/5
2	7	6	3	2	1	5/6
3	11	5	2	2	1	4/5
4	8	6	2	3	1	4/6
5	8	5	3	2	0	4/5
						21/27

Table 9 System performance on complex video sequences: for each sequence the number of tracks, action ground-truth ($\mathbf{W}_{\mathrm{GT}}$, $\mathbf{R}_{\mathrm{GT}}$,$\mathbf{O}_{\mathrm{GT}}$), and classification accuracy are reported.

Action	**Precision**	**Recall**
Walking	73%	85%
Running	77%	71%

Table 10 Precision and recall for the *running* and *walking* actions.

we used the videos containing a single person performing the same action multiple times.

Table 9 shows the performance of our approach on surveillance videos. For each sequence we report the detected tracks identified from our person detector and tracker. The tracks that contain less than 30 interest points are discarded and the filtered tracks are then used to perform action classification. These tracks are manually annotated in walking, running and other action (reported in table 9 in the columns $\mathbf{W}_{\mathrm{GT}}$, $\mathbf{R}_{\mathrm{GT}}$,$\mathbf{O}_{\mathrm{GT}}$ respectively). Details of classification accuracy are shown. We note that $21/27$ tracks are recognized correctly. The performance of action classification is evaluated in terms of two standard metrics i.e. precision and recall, defined as:

$$precision = \frac{\text{\# of correctly predicted actions}}{\text{\# of predicted actions}}, \tag{15}$$

$$recall = \frac{\text{\# of correctly predicted actions}}{\text{\# of ground-truth actions}}. \tag{16}$$

Precision and recall performance of the action recognition, also shown in table 10, are mostly affected by mistaken classification of the tracks that contain the "other" action, since only one track that contains a walking action was classified as running action.

Fig. 17 Example of two sequences from our multiple-action surveillance dataset. In the first sequence (seq. 3) our actors perform a *pickpocketing* event. In the second sequence (seq. 5) a *snatch* is performed.

6 Conclusions

In this chapter we have presented a novel method for human action categorization that exploits a new descriptor for spatio-temporal interest points that combines appearance (3D gradient descriptor) and motion (optic flow descriptor), and effective codebook creation based on radius-based clustering and a soft assignment of feature descriptors to codewords. The approach was validated on KTH and Weizmann datasets, on the Hollywood2 dataset and on a new surveillance dataset that contain unconstrained video sequences that include more realistic and complex actions. Results outperform the state-of-the-art with no parameter tuning. We have also shown that a strong reduction of computation time can be obtained by applying codebook size reduction with Deep Belief Networks, with small reduction of classification performance.

References

1. Arulampalam, M., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Transactions on Signal Processing **50**(2), 174–188 (2002)
2. Bagdanov, A.D., Dini, F., Del Bimbo, A., Nunziati, W.: Improving the robustness of particle filter-based visual trackers using online parameter adaptation. In: Proc. of AVSS (2007)
3. Ballan, L., Bertini, M., Del Bimbo, A., Seidenari, L., Serra, G.: Event detection and recognition for semantic annotation of video. Multimedia Tools and Applications **51**(1), 279–302 (2011)
4. Ballan, L., Bertini, M., Del Bimbo, A., Seidenari, L., Serra, G.: Effective codebooks for human action representation and classification in unconstrained videos. IEEE Transactions on Multimedia **14**(4), 1234–1245 (2012)
5. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: The CLEAR MOT metrics. EURASIP Journal on Image and Video Processing (2008)

6. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(3), 257–267 (2001)
7. Bregonzio, M., Gong, S., Xiang, T.: Recognising action as clouds of space-time interest points. In: Proc. of CVPR (2009)
8. Cao, L., Zicheng, L., Huang, T.: Cross-dataset action detection. In: Proc. of CVPR (2010)
9. Carreira Perpinan, M.A., Hinton, G.E.: On contrastive divergence learning. In: Proc. of AISTATS (2005)
10. Chen, M.Y., Hauptmann, A.G.: MoSIFT: Recognizing human actions in surveillance videos. Tech. rep., CMU (2009)
11. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(5), 603–619 (2002)
12. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. of CVPR (2005)
13. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: Proc. of VSPETS (2005)
14. Efros, A.A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: Proc. of ICCV (2003)
15. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: Proc. of CVPR (2003)
16. Gao, Z., Chen, M.Y., Hauptmann, A.G., Cai, A.: Comparing evaluation protocols on the KTH dataset. In: Proc. of HBU Workshop (2010)
17. van Gemert, J.C., Veenman, C.J., Smeulders, A.W.M., Geusebroek, J.M.: Visual word ambiguity. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(7), 1271–1283 (2010)
18. Gorelick, L., Blank, M., Schechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(12), 2247–2253 (2007)
19. Hauptmann, A.G., Christel, M.G., Yan, R.: Video retrieval based on semantic concepts. Proceedings of the IEEE **96**(4), 602–622 (2008)
20. Hinton, E.G., Osindero, S., Teh, Y.: A fast learning algorithm for deep belief nets. Neural Computation **18**(7), 1527–1554 (2006)
21. Hinton, E.G., Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
22. Jiang, Y.G., Yang, J., Ngo, C.W., Hauptmann, A.G.: Representations of keypoint-based semantic concept detection: a comprehensive study. IEEE Trans. on Multimedia **12**(1), 42–53 (2010)
23. Jurie, F., Triggs, B.: Creating efficient codebooks for visual recognition. In: Proc. of ICCV (2005)
24. Kläser, A., Marszałek, M., Schmid, C.: A spatio-temporal descriptor based on 3D-Gradients. In: Proc. of BMVC (2008)
25. Kong, Y., Zhang, X., Hu, W., Jia, Y.: Adaptive learning codebook for action recognition. Pattern Recognition Letters **32**(8), 1178–1186 (2011)
26. Kovashka, A., Grauman, K.: Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: Proc. of CVPR (2010)
27. Laptev, I.: On space-time interest points. International Journal of Computer Vision **64**(2-3), 107–123 (2005)
28. Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: Proc. of CVPR (2008)
29. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proc. of CVPR (2006)
30. Lin, Z., Jiang, Z., Davis, L.S.: Recognizing actions by shape-motion prototype trees. In: Proc. of ICCV (2009)
31. Liu, J., Ali, S., Shah, M.: Recognizing human actions using multiple features. In: Proc. of CVPR (2008)
32. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos "in the wild". In: Proc. of CVPR (2009)

33. Liu, J., Shah, M.: Learning human actions via information maximization. In: Proc. of CVPR (2008)
34. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. of DARPA IU Workshop (1981)
35. van der Maaten, L., Postma, E., van den Herik, H.: Dimensionality reduction: A comparative review. Tech. Rep. TiCC-TR 2009-005, Tilburg University (2009)
36. Marszałek, M., Laptev, I., Schmid, C.: Actions in context. In: Proc. of CVPR (2009)
37. Mikolajczyk, K., Leibe, B., Schiele, B.: Local features for object class recognition. In: Proc. of ICCV (2005)
38. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. International Journal of Computer Vision **65**(1/2), 43–72 (2005)
39. Mikolajczyk, K., Uemura, H.: Action recognition with motion-appearance vocabulary forest. In: Proc. of CVPR (2008)
40. Moeslund, T., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. Computer Vision and Image Understanding **104**(2-3), 90–126 (2006)
41. Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. International Journal of Computer Vision **79**(3), 299–318 (2008)
42. Poppe, R.: Vision-based human motion analysis: An overview. Computer Vision and Image Understanding **108**(1-2), 4–18 (2007)
43. Poppe, R.: A survey on vision-based human action recognition. Image and Vision Computing **28**(6), 976–990 (2010)
44. Rapantzikos, K., Avrithis, Y., Kollia, S.: Dense saliency-based spatiotemporal feature points for action recognition. In: Proc. of CVPR (2009)
45. Schüldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. In: Proc. of ICPR (2004)
46. Scovanner, P., Ali, S., Shah, M.: A 3-Dimensional SIFT descriptor and its application to action recognition. In: Proc. of ACM Multimedia (2007)
47. Shao, L., Gao, R., Liu, Y., Zhang, H.: Transform based spatio-temporal descriptors for human action recognition. Neurocomputing **74**(6), 962–973 (2011)
48. Shao, L., Mattivi, R.: Feature detector and descriptor evaluation in human action recognition. In: Proc. of CIVR (2010)
49. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proc. of ICCV (2003)
50. Snoek, C.G.M., Worring, M., van Gemert, J.C., Geusebroek, J.M., Smeulders, A.W.M.: The challenge problem for automated detection of 101 semantic concepts in multimedia. In: Proc. of ACM Multimedia (2006)
51. Sun, X., Chen, M., Hauptmann, A.G.: Action recognition via local descriptors and holistic features. In: Proc. of CVPR4HB Workshop (2009)
52. Turaga, P., Chellappa, R., Subrahmanian, V., Udrea, O.: Machine recognition of human activities: A survey. IEEE Transactions on Circuits and Systems for Video Technology **18**(11), 1473–1488 (2008)
53. Vezzani, R., Cucchiara, R.: Video surveillance online repository (ViSOR): an integrated framework. Multimedia Tools and Applications **50**(2), 359–380 (2010)
54. Wang, H., Ullah, M.M., Kläser, A., Laptev, I., Schmid, C.: Evaluation of local spatio-temporal features for action recognition. In: Proc. of BMVC (2009)
55. Wang, Y., Mori, G.: Max-margin hidden conditional random fields for human action recognition. In: Proc. of CVPR (2009)
56. Willems, G., Tuytelaars, T., Van Gool, L.: An efficient dense and scale-invariant spatio-temporal interest point detector. In: Proc. of ECCV (2008)
57. Wong, S.F., Cipolla, R.: Extracting spatiotemporal interest points using global information. In: Proc. of ICCV (2007)
58. Wu, B., Nevatia, R.: Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. International Journal of Computer Vision **75**(2), 247–266 (2007)

59. Yao, A., Gall, J., Van Gool, L.: A hough transform-based voting framework for action recognition. In: Proc. of CVPR (2010)
60. Yilmaz, A., Shah, M.: Actions sketch: a novel action representation. In: Proc. of CVPR (2005)
61. Yu, G., Goussies, N., Yuan, J., Liu, Z.: Fast action detection via discriminative random forest voting and top-k subvolume search. IEEE Transactions on Multimedia **13**(3), 507–517 (2011)
62. Zhang, J., Marszałek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. International Journal of Computer Vision **73**(2), 213–238 (2007)