Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

Fusing appearance and distribution information of interest points for action recognition

Matteo Bregonzio*, Tao Xiang, Shaogang Gong

School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, UK

ARTICLE INFO

Article history:
Received 23 February 2010
Received in revised form
26 July 2011
Accepted 12 August 2011

Keywords:
Action recognition
Clouds of Points
Feature fusion
Interest points detection
Multiple Kernel Learning

ABSTRACT

Most of the existing action recognition methods represent actions as bags of space-time interest points. Specifically, space-time interest points are detected from the video and described using appearance-based descriptors. Each descriptor is then classified as a video-word and a histogram of these video-words is used for recognition. These methods therefore rely solely on the discriminative power of individual local space-time descriptors, whilst ignoring the potentially useful information about the global spatio-temporal distribution of interest points. In this paper we propose a novel action representation method which differs significantly from the existing interest point based representation in that only the global distribution information of interest points is exploited. In particular, holistic features from clouds of interest points accumulated over multiple temporal scales are extracted. Since the proposed spatio-temporal distribution representation contains different but complementary information to the conventional Bag of Words representation, we formulate a feature fusion method based on Multiple Kernel Learning. Experiments using the KTH and WEIZMANN datasets demonstrate that our approach outperforms most existing methods, in particular under occlusion and changes in view angle, clothing, and carrying condition.

1. Introduction

Human action recognition in videos has recently become a very active research area. It is an important component in various application areas including: video surveillance, video indexing and browsing, recognition of gestures, human–computer interaction, and analysis of sport-events. Despite the best efforts of a large number of computer vision researchers, action recognition remains largely an unsolved problem. This is because in real world the same actions can be performed by subjects of different sizes, appearances, and poses. Moreover, actions can be captured under occlusion from static or moving objects, illumination changes, shadows, and camera movements.

Most of the previous approaches on action recognition focus on the action representation problem. Early work is based on tracking [30,1,29,32] or spatio-temporal shape templates [15,19,41]. Both tracking and spatio-temporal shape template construction require extraction of highly detailed silhouettes, which may not be possible given a real-world noisy video input. To address this problem, space-time interest point based approaches have become increasingly popular [9,7,26,31,25]. These approaches are based on a Bag of Words (BOW) feature representation that has been successfully applied to 2D object categorisation and detection [8]. Compared with tracking and spatio-temporal shape based approaches, they are more robust to noise, camera movement, and low-resolution inputs. Nevertheless, they rely solely on the discriminative power of individual local space-time descriptors. In other words, only the appearance information captured by each individual interest point is utilised; information about the global spatio-temporal distribution of interest points is ignored. Consequently, they are unable to capture smooth and fast motions due to the lack of large scale temporal information. Furthermore, they have to address the non-trivial problems of selecting an optimal space-time descriptor, clustering algorithm for constructing a codebook and codebook size, all of which inevitably involve parameter tuning. Such parameter settings are highly data dependent and re-tuning is required for different video inputs.

To address the limitations of the conventional BOW action representation method, we propose a novel approach based on representing action as clouds of interest points accumulated at different temporal scales. Specifically, we adopted a new space-time interest point detection method to extract denser and more informative interest points compared to existing interest point extraction methods [9,7]. In particular, our model avoids spurious detection in both background areas and highly textured static

* Corresponding author. Tel.: +44 20 7882 8019; fax: +44 20 8980 7064.
E-mail addresses: bregonzio@eecs.qmul.ac.uk (M. Bregonzio), txiang@eecs.qmul.ac.uk (T. Xiang), sgg@eecs.qmul.ac.uk (S. Gong).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.08.014

foreground areas unrepresentative of the dynamic parts of actions concerned. The extracted interest points are accumulated over time at different temporal scales to form point clouds. Examples of the clouds of interest points of different temporal scales are shown in Fig. 1. Holistic features are then computed from these point clouds for action representation, which capture *explicitly* and *globally* the spatial and temporal distribution of salient local space-time patches. Differing from the existing interest point based representation approaches, the proposed method extracts holistic and global information about action at multiple scales rather than local appearance information at each individual interest point.

Since Bag of Words (BOW) and the proposed Clouds of Points (COP) representations exploit completely different yet complementary aspects of an action, it is natural to fuse them to form a better representation. To this end, we formulate a feature fusion method based on Multiple Kernel Learning (MKL) [2]. Specifically, to learn a multi-class classifier for action recognition, we employ a support vector machine with multiple kernels, each of them being computed using either the BOW features or our COP features at a certain scale. Multiple kernel learning is then performed to learn the best linear combination of the kernels to yield the optimal classification accuracy. Learned in a one-vs-all setting, the multiple kernel SVM can automatically identify which type of features are the most informative ones in discriminating one specific class of actions from others.

The proposed approach is evaluated in depth on two widely used public datasets, namely the KTH dataset [7] and the Weizmann Institute of Science (WEIZMANN) dataset [15]. The experimental results demonstrate that our approach outperforms most of the existing methods. In particular, with only the proposed COP features, very competitive results can be obtained. When combined with the complementary BOW features, better performance can be achieved. Furthermore, we tested our method on the WEIZMANN robustness test dataset. The result suggests that the proposed approach is more robust against occlusion and changes in view angle, clothing, carrying condition compared to existing methods. In addition, we examine the performance of our method on the more challenging YouTube dataset [24], which features constant and unpredictable camera movements and dynamic background, in order to identify its limitations and suggest possible extensions.

2. Related work

Existing human action representation approaches can be broadly classified into four categories: flow based approaches [10], spatio-temporal shape template based approaches [15,19,41], interest points based approaches [9,7,26,31,23,14,43,35,17,22,34], and tracking based approaches [30,1,29,32]. Flow based approaches construct action templates based on optical flow computation [10,11]. However, the features extracted from flow templates are sensitive to noise, especially at the boundaries of the human body. Spatio-temporal shape template based approaches essentially treat the action recognition problem as a 3D object recognition problem by representing an action using features extracted from the spatio-temporal volume of an action sequence [15,19,41]. These approaches require highly detailed silhouettes to be extracted, which may not be possible given a real-world noisy video input. In addition, the computational cost of the space-time volume based approaches is unacceptable for real-time applications. Tracking based approaches [30,1,29,32] suffer from the same problems. Consequently, although 100% recognition rate has been reported on the 'clean' Weizmann Institute of Science (WEIZMANN) dataset, these approaches would not work well on noisy datasets such as the KTH dataset, which is featured with low resolution, strong shadows, and camera movement that renders clean silhouette extraction impossible.

To address this problem, Schüldt et al. [7] propose to represent action using 3D space-time interest points detected from video. The detected points are clustered to form a dictionary of prototypes or video-words. Each action sequence is then represented following the Bag of Words (BOW) paradigm. Dollar et al. [9] introduce a multidimensional linear filter detector, which results in the detection of denser interest points compared to alternative detectors such as 3D Harris detector [18]. However, as mentioned earlier, their method ignores the potential valuable information provided by the global spatio-temporal distribution of the interest points. Consequently, they are unable to capture smooth and fast motions due to the lack of temporal information. This also explains why they generate poor results on the clean yet more ambiguous WEIZMANN dataset whilst working reasonably well on the KTH dataset, compared with methods using holistic representation such as the spatio-temporal shape template [15,19,41].

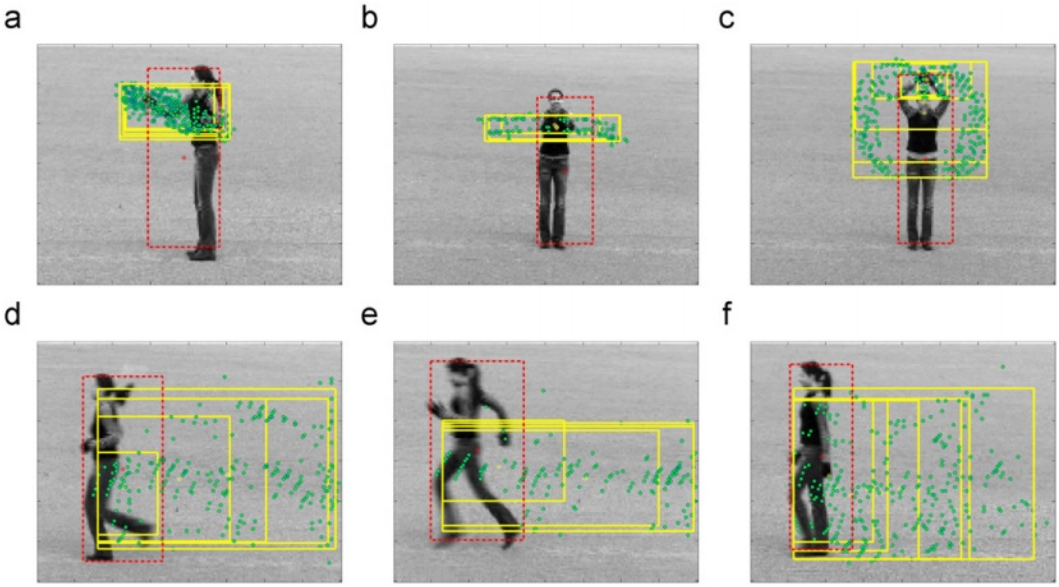


Fig. 1. Examples of clouds of interest points extracted from the KTH dataset [7]. The point clouds at different temporal scales are highlighted in yellow boxes. (a) Boxing, (b) clapping, (c) hand waving, (d) jogging, (e) running, (f) walking. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Please cite this article as: M. Bregonzio, et al., Fusing appearance and distribution information of interest points for action recognition, Pattern Recognition (2011), doi:10.1016/j.patcog.2011.08.014

A number of researchers have realised the limitations of the conventional BOW model and started to exploit the information about the spatial and temporal distribution of interest points. Liu and Shah exploit the spatial distribution of interest points using a modified correlogram [23]. Gilbert et al. [14] encode spatial information through concatenating video-words detected in different regions. Zhang et al. [43] introduce the concept of motion context to capture both spatial and temporal distributions of video-words. However, all these extensions still suffer from the same problem as the original BOW method, that is, they still have to go through the non-trivial process of selecting the optimal space-time descriptor, clustering algorithm for constructing a codebook, and codebook size. In addition, spatial and temporal information about the distribution of the interest points is only exploited implicitly, locally, and at a fixed temporal scale. In comparison, the proposed Clouds of Points (COP) representation exploits the distribution information explicitly and at multiple temporal scales, therefore capturing both local and global temporal information about the distribution.

There have also been a number of attempts on fusing features extracted for representing actions. Yao et al. [40] extract deformable templates constructed with flow and shape features and then use them to represent actions. The fusion is performed at the feature level through a weighted sum of flow and shape primitives. Similarly, Ikizler et al. [17] extract flow and shape features and then fuse them via a voting scheme. Instead of giving equal weight to different features, a weighted voting scheme is also tested and different combinations of weights are manually selected, but no improvement was obtained. An alternative fusion strategy is presented in [22], which builds a shape-motion prototype tree and classifies actions using prototype-matching. Feature fusion is performed by simple concatenation of shape-motion prototypes. Our approach differs from these methods in two aspects: (1) the two types of features, COP and BOW features are both extracted from space-time interests points, which are easy to compute, and robust against noise and occlusion. (2) The fusion method used in our work is based on multiple kernel learning (MKL) which automatically determines which features are more relevant for the classification task and assigns weights accordingly. Our results suggest that this leads to superior performance compared with simple concatenation. Note that MKL has recently been adopted to fuse three different context features for action recognition [34]. Compared with [34], our method does not rely on feature tracking and is computationally much more efficient.

The idea of extracting spatio-temporal distribution features was first exploited in our previous work [5]. Compared with [5], this work differs significantly in the following: (1) features extracted from clouds of interest points of multiple temporal scales are combined in a more principled manner via kernel methods. (2) The complementary nature of the proposed clouds of interest point features and bag of words features are exploited and a multiple kernel learning method is formulated to fuse them together, which leads to improvement on both recognition accuracy and robustness against occlusion.

3. Interest points detection

Space-time interest points are local spatio-temporal features considered to be salient or descriptive of actions captured in a video. Among various interest point detection methods, the one proposed by Dollar et al. [9] is perhaps the most widely used for action recognition. Using their detector, intensity variations in the temporal domain are detected using Gabor filtering. The detected interest points correspond to local 3D patches that undergo complex motions. Specifically, the response function of the Gabor filters has the following form:

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2 \tag{1}$$

where $g(x,y:\sigma)$ is the Gaussian smoothing kernel applied in the spatial domain, while h_{ev} and h_{od} are the 1D Gabor filters applied temporally, defined as:

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

$$h_{od}(t;\tau,\omega) = -\sin(2\pi t\omega)e^{-t^2/\tau^2} \tag{3}$$

As reported in the original paper [9], by setting $\omega = 4/\tau$, there are essentially two free parameters τ and σ which roughly control the spatial and temporal scales of the detector.

Despite its popularity, the Dollar detector has a number of drawbacks: it ignores pure translational motions, since it uses solely local information within a small region; it is prone to false detection due to video noise; it also tends to generate spurious detection in background areas surrounding object boundaries and in highly textured areas that are not in motion; it is particularly ineffective given slow object movement, small camera movement, or camera zooming. Some of those drawbacks are highlighted in the examples shown in Fig. 2.

A new interest point detector is developed here to overcome the shortcomings of the Dollar detector. In particular, most of the shortcomings of the Dollar detector are caused by its design of spatial and temporal filters and the way these filters are combined to give the final response. Especially, the 1D Gabor filter applied in the temporal domain is sensitive to background noise and highly textured background/foreground areas which have nothing to do with the action being performed. To overcome this problem, the proposed detector adopts different and more effective filters for detecting salient space-time local areas undergoing complex motions. More specifically, our interest point detection method consists of two steps: (1) frame differencing for focus of attention and region of interest detection[1]; and (2) Gabor filtering on the detected regions of interest using 2D Gabor filters of different orientations. Via these two steps, saliency detection in both the temporal and spatial domains are combined together to give the filter response.

The Gabor filters are applied on the frame difference result. Specifically, the 2D Gabor filters are composed of two parts. The first part $s(x,y;i)$ represents the real part of a complex sinusoid, known as the carrier:

$$s(x,y;i) = \cos(2\pi(\mu_0 x + \upsilon_0 y) + \theta_i) \tag{4}$$

where θ_i defines the orientation of the filter and eight orientations are considered:

$$\theta_{i=1,..,8} = \{0^\circ, \pm 22^\circ, \pm 45^\circ, \pm 67^\circ, 90^\circ\} \tag{5}$$

and μ_0 and υ_0 are the spatial frequencies of the sinusoid controlling the scale of the filter. The second part of the filter $G(x,y)$ represents a 2D Gaussian-shaped function, known as the envelope:

$$G(x,y) = \exp\left(\frac{-\frac{x^2}{\rho^2} + \frac{y^2}{\rho^2}}{2}\right) \tag{6}$$

where ρ is the parameter that controls the width of $G(x,y)$. We have $\mu_0 = \upsilon_0 = 1/2\rho$; therefore, the only parameter controlling the

[1] Although it is a very simple technique, frame differencing is found to be sufficient for our interest point detector given moderate camera motions such as those in the KTH dataset; When larger camera movements are present, a more sophisticated foreground detection method need to be adopted (e.g. one can employ an object detector such as [12]).

Please cite this article as: M. Bregonzio, et al., Fusing appearance and distribution information of interest points for action recognition, Pattern Recognition (2011), doi:10.1016/j.patcog.2011.08.014

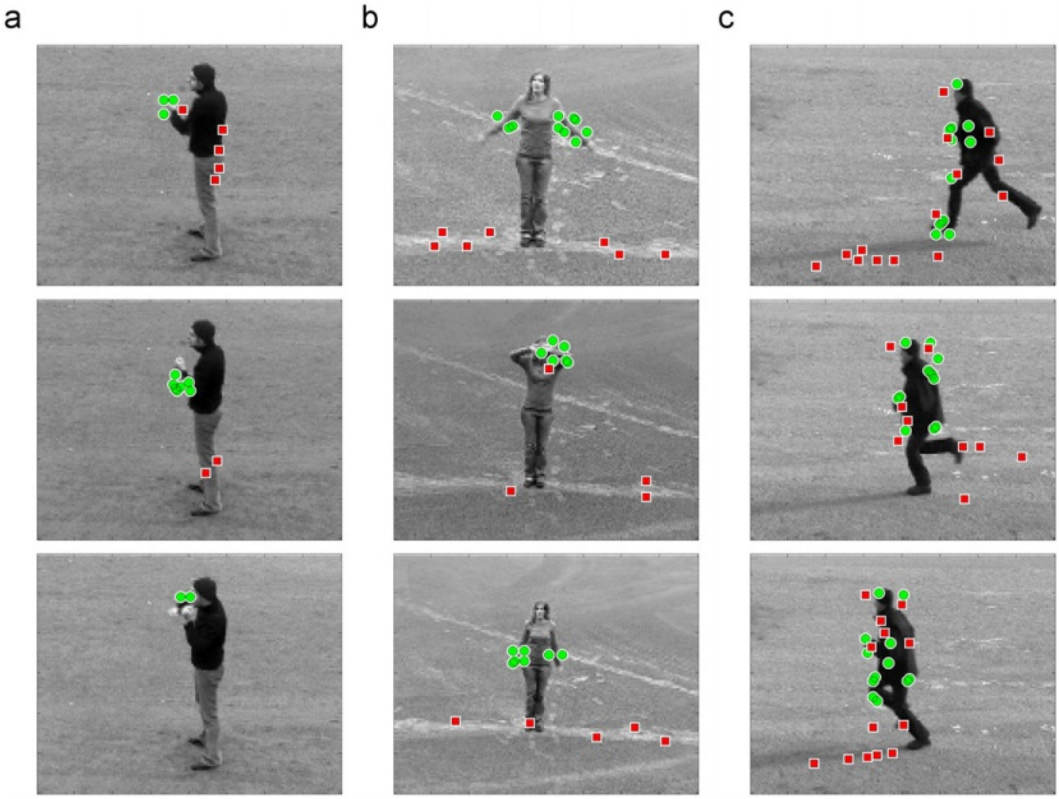


Fig. 2. Comparison between interest points detected using our detector (green circle points) and the Dollar et al. [9] detector (red square points). (a) Boxing, (b) hand waving, (c) running. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

scale is ρ, which is set to 11 pixels in this study.[2] The eight Gabor filters are applied separately and eight different responses are computed at each frame. Examples of 2D Gabor filters oriented along different directions are shown in Fig. 3. These responses are combined together to compute a bi-dimensional saliency map S_t^{map} as follows:

$$S_t^{map} = (S_t^{0^\circ})^2 + (S_t^{22^\circ})^2 + (S_t^{-22^\circ})^2 + (S_t^{45^\circ})^2 + (S_t^{-45^\circ})^2 + (S_t^{67^\circ})^2 + (S_t^{-67^\circ})^2 + (S_t^{90^\circ})^2 \quad (7)$$

where that image coordinates (x,y) are omitted for conciseness. Finally, interest points are detected as local maxima of the saliency map.

Fig. 2 shows examples of our interest point detection results obtained on the KTH dataset. It is evident that the detected interest points are much more meaningful and informative compared with those detected using the Dollar et al. [9] detector. In particular, the interest points detected by our approach tend to correspond to the main body parts contributing to the action being performed, whilst those detected by the Dollar detector often drift to static body parts or to background areas with strong edges. The experiments presented in Section 7.3 also suggest that a better recognition performance can be obtained when our interest point detector is used in place of the Dollar et al. [9] detector, either with the standard Bag of Words representation or the proposed Clouds of Points representation.

4. Action representation

4.1. Clouds of interest points

Consider an action video sequence $\mathbf{V}$ consisting of T image frames, represented as:

$$\mathbf{V} = [\mathbf{I}_1, \ldots, \mathbf{I}_t, \ldots, \mathbf{I}_T] \quad (8)$$

where $\mathbf{I}_t$ is the tth image frame. For the image frame $\mathbf{I}_t$, a total of S interest point clouds of different temporal scales are formed. They are denoted as $[\mathbf{C}_t^1, \ldots, \mathbf{C}_t^s, \ldots, \mathbf{C}_t^S]$. More specifically, an interest point cloud of the s-th scale is constructed by accumulating the interest points detected over the past $s \times N_s$ frames, where N_s is the difference between two consecutive scales (in the number of frames). Examples of clouds of interest points formed using the KTH and WEIZMANN datasets are shown in Fig. 4. It can be seen from Fig. 4 that different types of actions result in interest point clouds of very different shapes, relative locations (w.r.t. body location), and distributions. It is also evident that interest point clouds of different scales capture different aspects of human motion that potentially have different levels of discriminative power. This will be exploited by the feature selection method detailed later (Section 4.3).

4.2. Feature extraction

For the S interest point clouds constructed for the t-th image frame $[\mathbf{C}_t^1, \ldots, \mathbf{C}_t^s, \ldots, \mathbf{C}_t^S]$, two sets of features are extracted. These features are significantly different from the local descriptors computed by conventional interest point based approaches. In particular, the interest point cloud features are global and holistic capturing distribution information of interest points, whilst the

[2] The value of ρ is set empirically. It could be set in a more principal way via cross validation. It has been observed in our experiments that the recognition performance is not sensitive to the value of ρ.

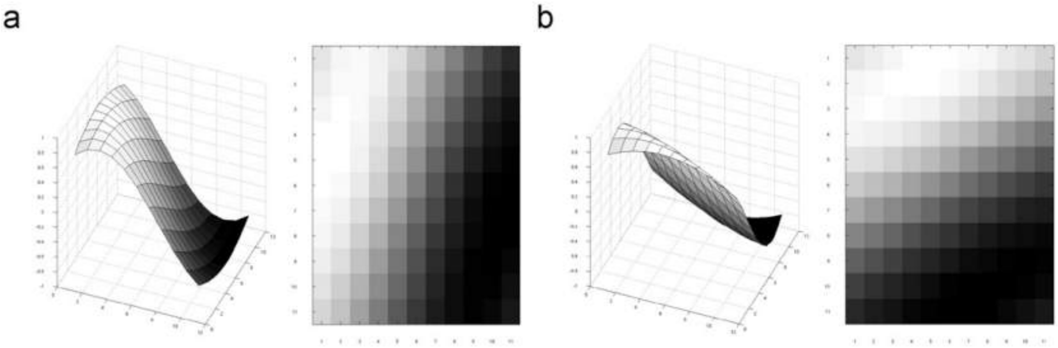


Fig. 3. Examples of the 2D Gabor filters oriented along (a) 22° and (b) 67°.

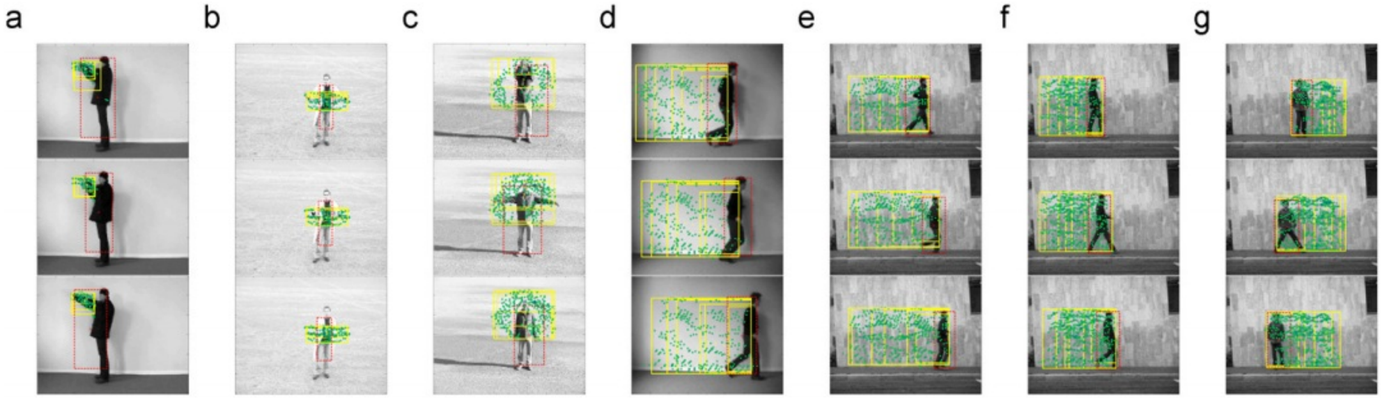


Fig. 4. Examples of clouds of space-time interest points. We have $S=6$ and $N_s=5$. In each frame the red rectangle represents the foreground area, the green points are the extracted interest points, and the yellow rectangles illustrate clouds of different scales. (a) Boxing, (b) hand clapping, (c) hand waving, (d) jogging, (e) running, (f) walking, (g) galloping sideways. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

conventional descriptor features, computed from a cuboid centred at each interest point are local, describing appearance information of individual interest points.

The first set of interest point cloud features is concerned with the shape and speed of foreground objects. To reliably detect and segment a foreground object given camera movement, zooming, strong shadows, and noisy input is a non-trivial task. This is accomplished by the following procedure. First, regions of interest are detected via frame difference. Second, a series of 2D Gabor filters are applied to the image frame. Third, the responses of these filters are fused together with the frame difference result. Finally, a Prewitt edge detector [27] is employed to segment the object from the detected foreground area. Once an object is segmented from the frame, two features are computed from each frame: O_t^r measuring the height and width ratio of the object, and O_t^{Sp} measuring the absolute speed of the object measured in pixels per frame ($O_t^{Sp}=0$ means no object displacement).

The second set of features are extracted from interest point clouds of different scales, they are thus scale dependent. Particularly, from the s-th scale cloud, eight features are computed and denoted as

$$[C_s^r, C_s^{Sp}, C_s^D, C_s^{Vd}, C_s^{Hd}, C_s^{Hr}, C_s^{Wr}, C_s^{Or}] \tag{9}$$

Note that subscript t is omitted for clarity. Specifically, C_s^r is the height and width ratio of the cloud; C_s^{Sp} is the absolute speed of the cloud measured in pixels per frame ($C_s^{Sp}=0$ means no cloud movement); C_s^D is the density of the interest point within the cloud, which is computed as the total number of points normalised by the area of the cloud ($C_s^D=1$ means 1 point detected in a 10×10 pixels cloud area); C_s^{Vd} and C_s^{Hd} measure the spatial relationship between the cloud and the detected object area measured in pixels. Specifically, C_s^{Vd} is the vertical distance between the geometrical centre (centroid) of the object area and the cloud, and C_s^{Hd} is the distance in the horizontal direction ($C_s^{Vd}=0$ and $C_s^{Hd}=0$ means that the cloud and the object are centred in the same place). C_s^{Hr} and C_s^{Wr} are the height ratio and width ratio between the object area and the cloud respectively ($C_s^{Hr}=1$ and $C_s^{Wr}=1$ mean that object and cloud have the same size). C_s^{Or} measures how much the two areas overlap in terms of percentage. Overall, the eight features can be put into two categories: C_s^r, C_s^{Sp}, and C_s^D measure the shape, speed, and density of the cloud itself; the five remaining features capture the relative shape and location information between the object and the cloud areas. To make these features insensitive to outliers in the detected interest points, an outlier filter is deployed before the feature extraction, which evaluates the interest point distribution over four consecutive frames and removes those points that are too far away from the distribution centroid. Specifically, we estimate the centroid of the points in each frame and compute the average distance from each point to the centroid. If the distance between an interest point and the centroid is four times or more of the average distance, it is most likely to be caused by background noise and thus removed.

Now each frame is represented using $8S+2$ features where S is the total number of scales (i.e. eight features for each scale plus two scale-independent features O_t^r and O_t^{Sp}). By using a total of $(8S+2)\ \times T$ features to represent the whole action sequence of T frames leads to a feature space of an extremely high dimension. It is well known that a high dimensional feature space can cause over-fitting resulting in poor recognition performance. To reduce the dimensionality of the feature space, and more importantly, to make our representation less sensitive to feature noise and invariant to the length of each action sequence, a histogram of N_b bins is constructed for each of the $8S+2$ features collected over

Please cite this article as: M. Bregonzio, et al., Fusing appearance and distribution information of interest points for action recognition, Pattern Recognition (2011), doi:10.1016/j.patcog.2011.08.014

time via linear quantization. Consequently, each action sequence is represented as $8S+2$ histograms or $(8S+2)\times N_b$ scalar features with $N_b \ll T$. Instead of using fixed-width histogram binning as most existing work does, we adopt histogram of non-uniform bin width with more bins being given to the high density area of the feature space [21]. More specifically, it involves the following steps: (1) For each feature, its values from different data points are represented as a random variable with a certain value range. (2) The Kaplan–Meier estimate of the cumulative distribution function (cdf), also known as the empirical cdf, is computed. (3) Plot the function with the random variable value as the x-axis and the cdf value (ranged from 0 to 1) as the y-axis. (4) To build a N_b-bin histogram, draw N_b horizontal lines at equal interval along the y-axis. (5) The x coordinates of the intersection points with the cdf plot determine the value range of each bin.

4.3. Feature selection

Using the $(8S+2)\times N_b$ features as described above, the feature space dimension is still very high and needs to be further reduced. Moreover, there are uninformative and redundant features one would wish to eliminate from the feature set. To that end, a simple and intuitive yet effective feature selection method is formulated below.

Our feature selection method measures the relevance of each feature according to how much its value varies within each action class and across different classes. Specifically, a feature is deemed as being informative and relevant to the recognition task if its value varies little for actions of the same class but varies significantly for actions of different classes. First, given a training set of A action classes, for the i-th feature f_i in the a-th class, we compute its mean and standard deviation within the class as $\mu^a_{f_i}$ and $\sigma^a_{f_i}$ respectively. The relevant measure for feature f_i is then denoted as R_{f_i} and computed as:

$$\frac{R_{f_i}=\sqrt{\frac{1}{A}\sum_{a=1}^{A}(\mu^a_{f_i}-\hat{\mu_{f_i}})^2}}{\frac{1}{A}\sum_{a=1}^{A}\sigma^a_{f_i}} \tag{10}$$

where $\hat{\mu_{f_i}}=(1/A)\sum_{a=1}^{A}\mu^a_{f_i}$ is the inter-class mean of the A intra-class feature means. The numerator and denominator of the above equation correspond to the standard deviation of the intra-class means, and the inter-class mean of the intra-class standard deviations respectively. The former measures how the feature value varies across different classes (the higher the value is, the more informative the feature f_i is); the latter tells us how the value varies within each class (the lower the value, the more informative the feature). Overall, features with higher R_{f_i} values are preferred over those with lower ones. Finally, all features are ranked according to their R_{f_i} and a decision is made as to how many percent of the features are to be kept for recognition.

Our feature selection method, although intuitive, seems to have a number of drawbacks. First, different features are selected separately as if they were independent of each other. It has been widely recognised that combining good features together does not guarantee good recognition performance [28]. So, ideally we would like to select the features collectively. However, this means that the feature search space is too high for an exhaustive search and even a sequential-search based approximation scheme is considerably expensive. Second, more sophisticated relevance measures such as mutual information [28] can be used. Nevertheless, compared with alternative feature selection approaches, one of the advantages of our method is that it has an extremely low computational cost. We also show empirically through experiments (see Section 7.3) that our method is more effective than a far more complicated state-of-the-art method [28].

5. Combining multi-scale clouds of interest point features

The Clouds of Interest Points (COP) features are of multiple (S) temporal scales. Features of different scales may not be equally informative in representing different actions. This is because each class of actions have a specific temporal scale. In particular, different actions are performed by moving body parts at different speed. Most actions are periodic (e.g. running, walking, hand-clapping) consisting of repetitive cycles. For them, the lengths of their cycles are direct indications of their temporal scales. For instance, in the KTH dataset at 25 Hz, a full cycle of the running, hand-clapping and walking actions lasts around 20, 25, and 30 frames respectively. Intuitively, longer scale COP features are more useful in describing longer scale (slower) actions. Therefore it is necessary to weight the features of different scales according to their relevance to the classification task, and different weightings should be used for classifying different actions. Ideally these weightings should be learned automatically from a training dataset.

To this end, a multiple kernel learning (MKL) method is formulated for learning the optimal weighting of COP features of different scales for multi-class action classification. MKL was first introduced in [2] to address the problem of selecting the optimal combination of kernel functions for a specific feature for Support Vector Machine (SVM) classification. Recently it has been used in computer vision for addressing a closely related problem, that is, given a specific kernel function but different features capturing different aspects of a visual object, how to best combine them together to achieve the optimal classification performance [13,34]. In this work, we consider that COP features of different scales capture the characteristics of an action class under multiple temporal scales and MKL is adopted to learn the optimal combination of these features.

Let us formally define the multiple class action recognition problem. Taking the one-vs-rest scheme, C binary classifiers are learned to classify an action sequence into one of the C classes. Assume we have a training set $(x_i,y_i)_{i=1,\ldots,N}$ of N instances; each training sample x_i is a video sequence containing an action with a class label y_i. To represent the action, S features are extracted as described in Section 4. Each feature is a histogram corresponding to COP features at one specific scale. We denote the s-th scale feature as $f_s(x)$ where $f_s(\cdot)$ is the feature extraction function. Using multiple kernel learning, a set of kernel functions is to be computed, each of which is essentially a distance/similarity measure. Specifically, a kernel function

$$k_s(x,x')=k(f_s(x),f_s(x')) \tag{11}$$

measures the similarity between a pair of action sequences represented using the s-th scale COP features. For notational convenience, given an action sequence x, we denote its kernel response of the s-th feature to all N training samples as

$$K_s(x)=[k_s(x,x_1),k_s(x,x_2),\ldots,k_s(x,x_N)]^T \tag{12}$$

Now let us describe how different kernels corresponding to different COP features are combined in an SVM framework. Using MKL, the objective is to learn an optimal weighting so that the combined kernel function has the following form:

$$k^*(x,x')=\sum_{s=1}^{S}\beta_s k_s(x,x') \tag{13}$$

where β_s is the weight associated to the s-th temporal scale. To learn an SVM for classifying one action class against the rest, we have an optimisation problem with the following objective

Please cite this article as: M. Bregonzio, et al., Fusing appearance and distribution information of interest points for action recognition, Pattern Recognition (2011), doi:10.1016/j.patcog.2011.08.014

function:

$$\min_{\alpha,\beta,b} \frac{1}{2}\sum_{s=1}^{S} \beta_s \alpha^T K_s \alpha + C\sum_{i=1}^{N} L\left(y_i, b + \sum_{s=1}^{S} \beta_s K_s(x)^T \alpha\right)$$
$$\text{s.t.} \sum_{s=1}^{S} \beta_s = 1, \quad \beta_s \geq 0, \quad s = 1,\ldots,S \tag{14}$$

where α is a N-dimensional feature vector which can be seen as the weights of each training sample, b has a scalar value, K_s is defined in Eq. (12), $L(y,z)$ denotes the Hinge Loss function [3]. The two constraints put on β_s are to make sure that the estimated value of β_s is sparse and interpretable (i.e. as weights, they should be either zero or a positive number, and the sum of all weights should be 1). Various methods can be used to solve the above optimisation problem. In this work we adopted the semi-infinite linear program (SILP) algorithm [33]. Conventionally the multiple kernel learning problem is formulated as a convex quadratically constrained quadratic program and solved using a local descent algorithm such as Sequential Minimization Optimization (SMO). However, it is slow and only feasible for small scale problems. The method in [33] reformulates the multiple kernel learning problem as a semi-infinite linear program (SLIP), which can be efficiently solved using an off-the-shelf linear program solver and a standard SVM implementation. Two linear program solvers are formulated in [33]; one of them is a wrapper algorithm and the other a chunking algorithm. The wrapper algorithm was used in our implementation. Note that the regularisation constant C is determined via cross validation. Given the learned parameters β_s, α, and b, the final binary decision function of MKL is of the following form:

$$F_{MKL}(x) = \text{sign}\left(\sum_{s=1}^{S} \beta_s (K_s(x)^T \alpha + b)\right) \tag{15}$$

where sign$(\cdot)$ is a function that returns a value 1 if its parameter is positive and -1 if otherwise, $K_s(x)$ is defined in Eq. (12) which measures the similarity between the test data x with all N training data samples (both positive and negative). If $F_{MKL}(x)$ assumes the value 1, the test sequence x is deemed as being a member of the target action classes for which the MKL binary classifier is trained. Since we are dealing with a multiple class classification problem, multiple binary classifier is trained and a test action sequence is classified as the action class with the highest value of $\sum_{s=1}^{S} \beta_s (K_s(x)^T \alpha + b)$.

6. Appearance and distribution feature fusion

The proposed multi-scale COP features are fused with the conventional BOW interest point features to form the final representation of actions in our approach. This is because these two types of features capture completely different yet complementary aspects of actions: the former contains global distribution information of interest points, whilst the latter represents how each interest point looks like in terms of 3D texture and localised motion characteristics.

This fusion problem can be considered as a feature combination problem and addressed using the same multiple kernel learning method described in the preceding section. More specifically, after interest points are extracted using the method described in Section 3 and represented as a histogram of the BOW features, we compute a kernel function denoted as k_B and form a linear combination with the S COP features. Now the combined kernel function in Eq. (13) is rewritten as

$$k^*(x,x') = \sum_{s=1}^{S} \beta_s k_s(x,x') + \beta_B k_B(x,x') \tag{16}$$

Similarly, the objective function to be optimised using the SLIP algorithm [33] becomes

$$\min_{\alpha,\beta,b} \frac{1}{2}\left(\sum_{s=1}^{S} \beta_s \alpha^T K_s \alpha + \beta_B \alpha^T K_B \alpha\right)$$
$$+ C\sum_{i=1}^{N} L\left(y_i, b + \sum_{s=1}^{S} \beta_s K_s(x)^T \alpha + \beta_B K_B(x)^T \alpha\right)$$
$$\text{s.t.} \sum_{s=1}^{S} \beta_s + \beta_B = 1, \quad \beta_s \geq 0, \quad s = 1,\ldots,S, \quad \beta_B \geq 0 \tag{17}$$

where β_B is the weight of the BOW features. After parameter estimation, the final binary decision function is

$$F_{MKL}(x) = \text{sign}\left(\sum_{s=1}^{S} \beta_s (K_s(x)^T \alpha + b) + \beta_B (K_B(x)^T \alpha + b)\right) \tag{18}$$

7. Experiments

7.1. Datasets

KTH Dataset—The KTH dataset was provided by Schuldt et al. [7] in 2004 and is one of the largest public human activity video dataset. It contains six types of actions (boxing, hand clapping, hand waving, jogging, running and walking) performed by 25 subjects in four different scenarios including indoor, outdoor, changes in clothing and variations in scale. Each video clip contains one subject performing a single action. Each subject is captured in a total of 23 or 24 clips, giving a total of 599 video clips. Each clip has a frame rate of 25 Hz and lasts between 10 and 15 s. The size of each image frame is 160 by 120 pixels. Examples of the KTH dataset are shown in Fig. 5.

WEIZMANN Dataset—The WEIZMANN dataset was introduced by Blank et al. [4] in 2005. It contains 90 video clips from nine different subjects. Again, each video clip contains one subject performing a single action. There are 10 different action categories: walking, running, jumping, galloping sideways, bending, one-hand-waving, two-hands-waving, jumping in place, jumping jack, and skipping. Each clip lasts about 2 s at 25 Hz. The image size is 180 by 144 pixels.

The same WEIZMANN group also provides a robustness test dataset. It includes 11 walking sequences with partial occlusions and non-rigid deformations (e.g. walking in skirt, walking with a briefcase, knees up walking, limping man, occluded legs, walking swinging a bag, sleepwalking, and walking with a dog). The dataset also includes nine walking sequences captured from different viewpoints (from 0° to 81° with 9° increments from the horizontal plane). This dataset is ideal for testing the robustness of an action recognition approach under occlusions, different views, and non-rigid deformations. Examples of the two WEIZMANN datasets can be seen in Fig. 5.

7.2. Experimental settings

All results were obtained using Leave-One-Out Cross-Validation (LOOCV) unless otherwise stated. It involved employing a group of clips from a single subject in a dataset as the testing data and the remaining clips as the training data. This was repeated so that each group of clips in the dataset is used once as the testing data. More specifically, for the KTH dataset, the clips of 24 subjects were used for training and the clips of the remaining subject was used for testing. For the WEIZMANN action recognition dataset, the training set contains eight subjects. As for the WEIZMANN robustness test dataset, the whole WEIZMANN action recognition dataset was used as training set, and each of the 20 robustness test sequences were classified as one of the 10 action classes.

Please cite this article as: M. Bregonzio, et al., Fusing appearance and distribution information of interest points for action recognition, Pattern Recognition (2011), doi:10.1016/j.patcog.2011.08.014

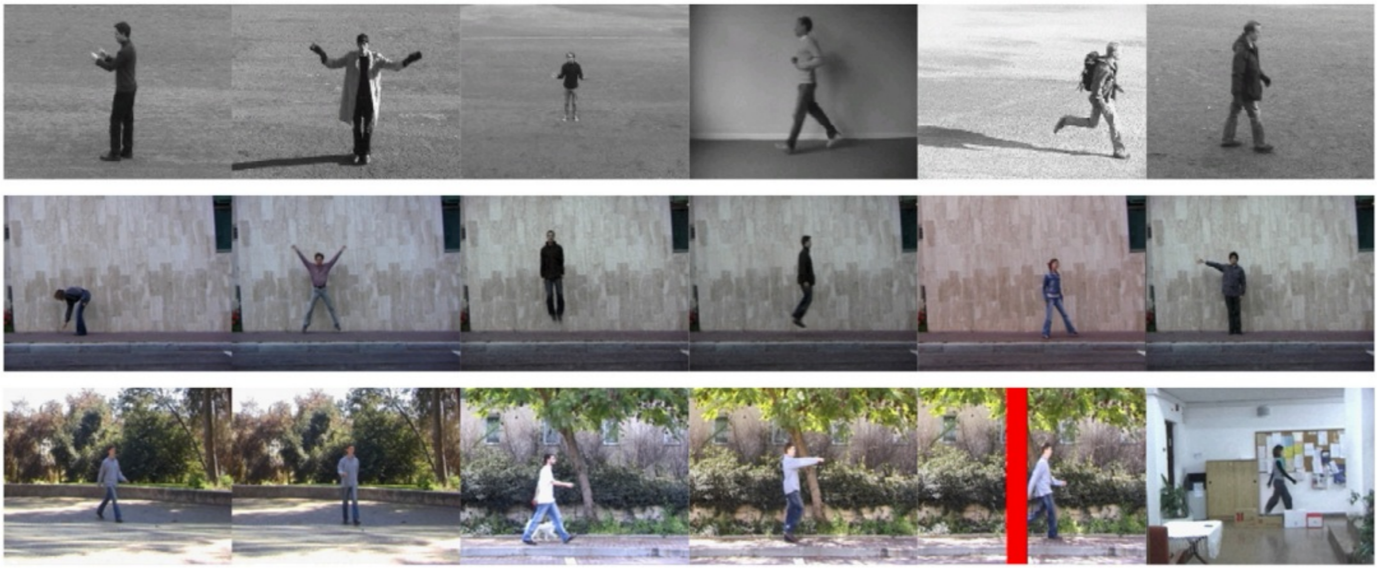


Fig. 5. From top to bottom: example frames from KTH dataset, WEIZMANN dataset and WEIZMANN robustness test dataset.

Table 1
Performance comparison between COP and BOW representations.

Dataset	BOW (%)	COP (%)
KTH	85.33	**92.83**
WEIZMANN	90.00	**96.66**

Table 2
Performance comparison between MKL and concatenation based feature combination.

Dataset	Concatenation (%)	MKL (%)
KTH	92.50	**92.83**
WEIZMANN	95.55	**96.66**

For constructing the multi-scale interest point clouds, the difference between two consecutive scales N_s was set to five frames and the total number of scales S was set to six. This gave us 50 features ($8S+2$), each of which was represented as a 50-bin histogram (i.e. our COP features were represented in a 2500-dimensional space). Twenty percent of these features was removed using our feature selection method (See Section 4.3).

For extracting the BOW features after interest points have been detected using our detector, we used the on-line available toolbox[3] implemented by Dollar [9] with the default setting. Specifically, the 3D Gradients descriptor was adopted to represent each interest point and a codebook was constructed by clustering the descriptors. The codebook size of 300 was used for KTH and 250 for WEIZMANN. Note that the codebook was constructed using a k-means clustering algorithm, which is sensitive to initialisation. Therefore, results are reported as an average of 20 trials. For the proposed COP features, no such initialisation issue exists, and different trials will give identical results. For the formulated MKL classifier, Gaussian kernels with a width of three were used. This parameter, as well as all other SVM parameters, was determined automatically through cross validation.

7.3. Recognition performance evaluation

Clouds of Points (COP) vs Bag of Words (BOW)—We compared the proposed COP with BOW representation. The recognition results are presented in the form of average recognition rates in Table 1 and confusion matrices in Fig. 7. Table 1 shows that the COP representation achieves higher average recognition rate on both datasets. Fig. 7 gives details on where the performance gain was obtained. It is noted that the COP representation is particularly strong in recognising jogging, running, and walking in the KTH dataset (comparing Fig. 7(b) with (a)), and running, skipping, and walking in the WEIZMANN dataset (comparing Fig. 7(e) with (d)). These actions are similar in terms of shape and motion appearance, but differ in terms of action speed and temporal evolution which can be measured globally and over different temporal scales. The BOW representation, based on interest point appearance only, is unable to capture these differences effectively; thus its performance is inferior. On the contrary, the proposed COP representation measures explicitly and globally the spatial and temporal distribution information. Moreover, it describes actions over multiple temporal scales,

Multi-scale recognition: MKL vs concatenation—Our COP representation contains features of multiple scales. Experiments were carried out to compare two ways of combining these multi-scale features: the proposed MKL method and the simple concatenation method. The former learns the optimal weighting from a training dataset, whilst the latter gives an equal weight to different feature scales. The obtained result, as shown in Table 2, indicates that MKL compares favourably with concatenation for combining COP features of different scales.

Fig. 6 shows the weight distributions over the multiple scale COP features learned by MKL. It can be seen that different weights are assigned to COP features of different scales, and the weight distributions vary for different actions. In particular, the results show that the learned weights are affected by the temporal scales of different actions. Let us look at the weight of the same features across different action classes in Fig. 6. It is noted that the weight for a longer scale feature is smaller for a faster action. This is because that for representing faster actions, long scale features become less informative than they are for slower actions. For instance, for the KTH dataset (Fig. 6(a)) the weight for the 30 frame scale COP feature is decreased when the action changes from walking to jogging, then

[3] http://vision.ucsd.edu/~pdollar/research/research.html.

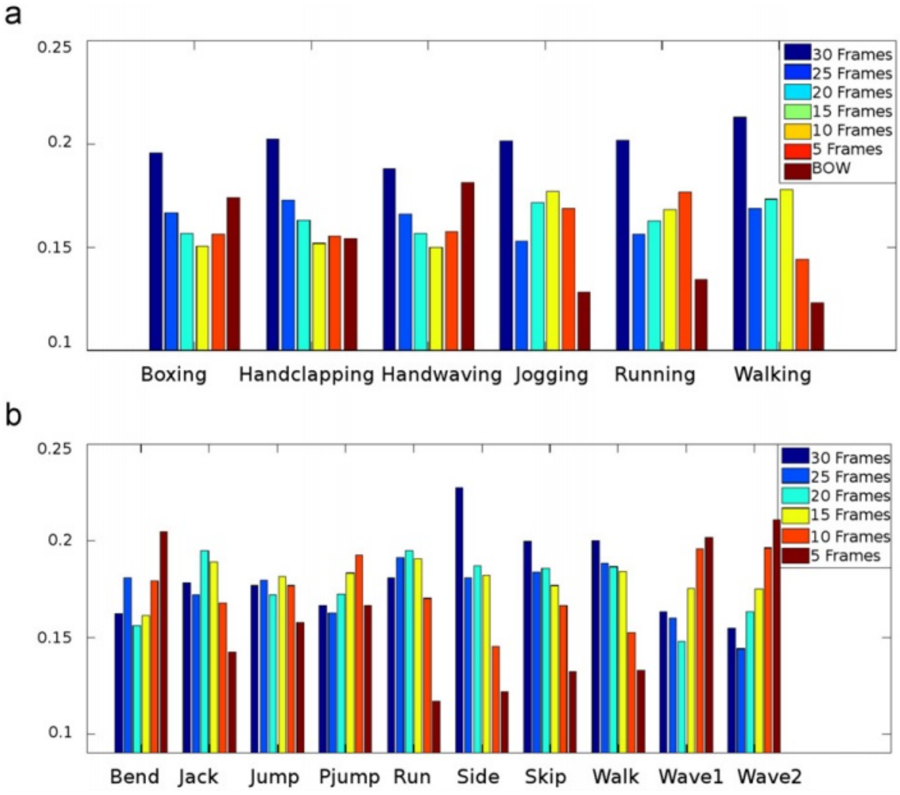


Fig. 6. Weight distribution of six multi-scale COP features learned using MKL. (a) KTH dataset. (b) WEIZMANN dataset.

Table 3
Effect of feature fusion.

Dataset	BOW (%)	COP (%)	Concatenation (%)	MKL fusion (%)
KTH	85.33	92.83	92.66	**94.33**
WEIZMANN	90.00	96.66	94.44	**96.66**

to running (i.e. getting faster). In the meantime, the short scale features have the opposite trend—the 5/10 frame scale COP features receive larger weight for running compared to jogging and walking. A similar trend can be observed for the WEIZMANN dataset. For example, Fig. 6(b) shows that the 30 frame COP feature (long scale) receives smaller weight for running than walking because running is faster than walking. By exploiting the different discriminative power of different feature scales, our MKL based feature combination is able to produce superior recognition performance than simple feature concatenation.

Effectiveness of feature fusion—Table 3 and Fig. 7 present a performance comparison between using a single type of features, either BOW and COP, and the fusion of them using MKL. Table 3 shows that an improvement is obtained by fusing the two complementary features together on the KTH dataset. Specifically, Fig. 7 shows that with feature fusion, the recognition rates for all six classes except handwaving were increased. It can also be seen in Table 3 that a simple concatenation based fusion has a negative effect on the recognition performance on both datasets.

Fig. 8 shows that different weight distributions were learned using MKL for different action classes. In general, the weights given to the BOW features are higher than those given to each single scale COP features, although overall more weights were given to COP features. It is interesting to note that the weighting distribution seems to be related to the temporal scale of different actions. In particular, for the KTH dataset, BOW features are given less weight for actions with shorter temporal scales (faster). For instance, Fig. 8(a) shows that the weights of BOW features for walking, jogging and running are descending in that order as the action gets faster. This is because when an action is performed with high motion intensity the computation of local appearance descriptors become unreliable, which decreases the discriminative power of the BOW features. However, due to factors such as noise, outliers and other non-linearities in the dataset, it is difficult to establish a clear trend on the feature weighting learned by MKL.

Table 4 also compares our results with the existing approaches proposed recently, which are not restricted to interest points based methods. It shows that our results are close to the best results reported so far on each dataset, and outperform most of the recently proposed methods, especially those tested on both datasets.

Interest point detector evaluation—The proposed interest point detector (Section 3) was initially compared with the widely used Dollar et al. [9] detector and the result is shown in Table 5.[4] As can be seen, with the same representation (COP and BOW fusion), our detector outperforms the Dollar et al. [9] detector on both the KTH and WEIZMANN datasets. This is because our detector is less sensitive to dynamic backgrounds and camera movements. Moreover, it tends to select more meaningful points located near the moving body parts (see Fig. 2). The improvement is particularly significant for the KTH dataset where dynamic background and camera motions appear frequently. Note that our detector differs from the Dollar et al. [9] detector in both the way the Gabor filters are designed and the use of frame differencing as a pre-processing step. To investigate the effect of each difference individually we also applied frame differencing to the Dollar et al. [9] detector. The result in Table 5 shows that an improvement can be obtained.

[4] Note that to obtain the Dollar et al. [9] detector results in Table 5, we used the same detector as in [9], but the representation after interest point detection is different.

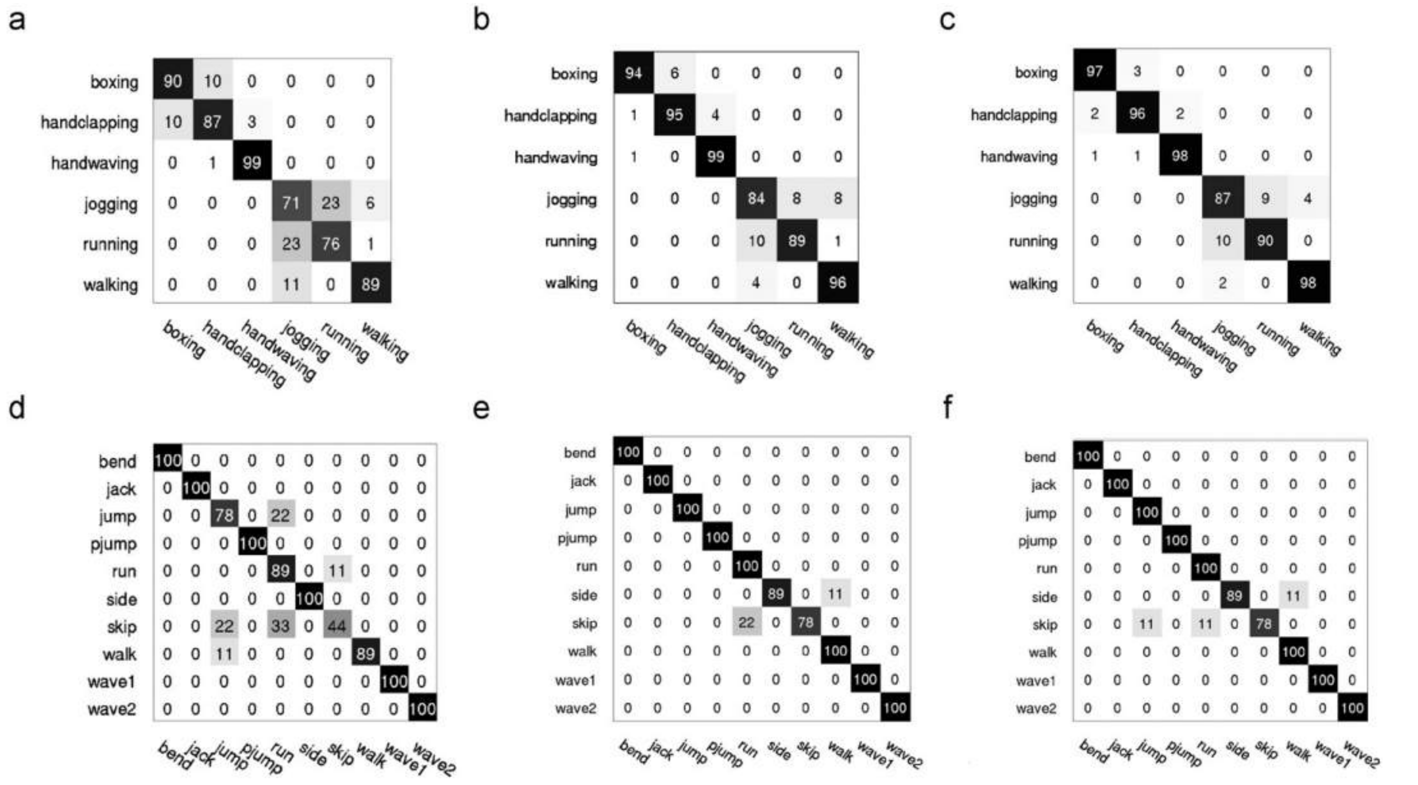


Fig. 7. Recognition performance measured using confusion matrices: (a) KTH dataset, BOW representation, accuracy: 85.33%; (b) KTH dataset, COP, accuracy: 92.83%; (c) KTH dataset, BOW+COP, accuracy: 94.33%; (d) WEIZMANN dataset, BOW representation, accuracy: 90.00%; (e) WEIZMANN dataset, COP, accuracy: 96.66%; (f) WEIZMANN dataset, BOW+COP, accuracy: 96.66%.

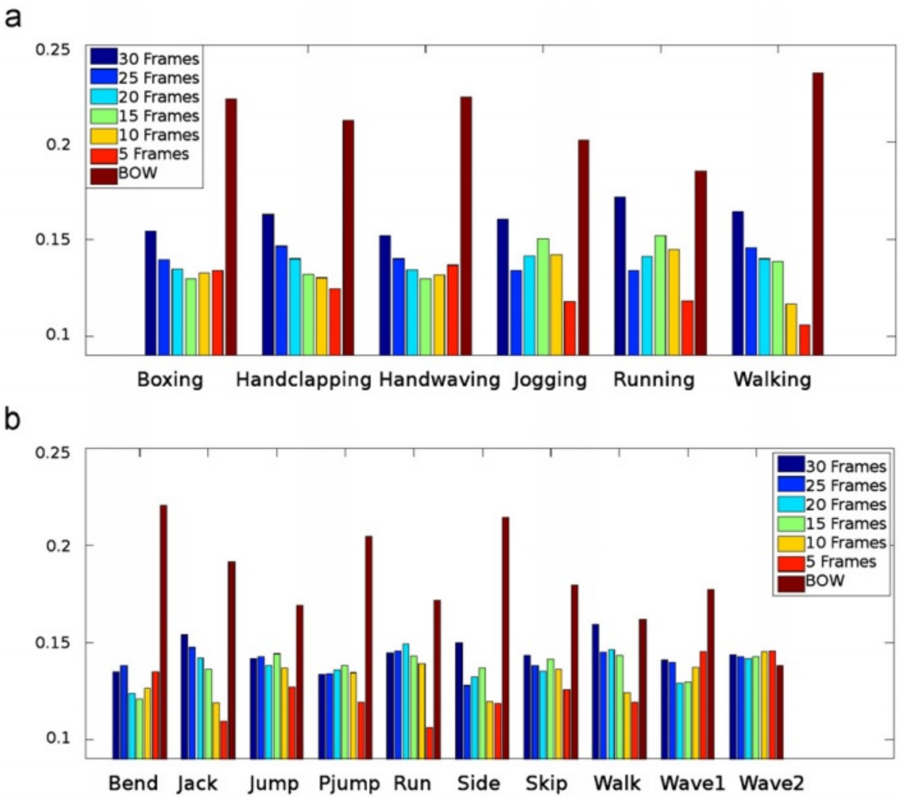


Fig. 8. Weight distribution between BOW and COP features. (a) KTH dataset and (b) WEIZMANN dataset.

However, the result is still worse than that of our detector. This suggests that the advantage of our detector is due to both the use of frame differencing and the way the Gabor filters are designed.

We also compared our detector with another popular detector, the 3D Harris interest point detector proposed by Laptev [18] which is a 3D extension of the Harris corner detector. In a recent

Table 4
Comparison with other methods. All methods used Leave-One-Out Cross-Validation (LOOCV), i.e. the same setting used in our experiments.

Method	KTH (%)	WEIZMANN (%)
Our approach	**94.33**	**96.66**
Sun et al. [35]	94.00	97.80
Ikizler et al. [17]	94.00	–
Lin et al. [22]	93.43	–
Wang et al. [38]	92.51	100
Lui [25]	92.30	–
Kläser et al. [20]	91.40	84.30
Niebles et al. [26]	83.30	90.00
Dollar et al. [9]	81.17	85.20
Liu et al. [23]	94.16	–
Zhao et al. [44]	91.17	–
Savarese et al. [31]	86.83	–

Table 5
Performance comparison between our interest point detector and the one presented by Dollar et al. [9] with and without frame differencing (FD). The experiment uses the same COP and BOW fusion representation with three different point detectors.

Dataset	Dollar et al. [9] detector (%)	Dollar et al. [9] detector with FD (%)	Our detector (%)
KTH	90.20	92.83	**94.33**
WEIZMANN	93.33	95.55	**96.66**

Table 6
Performance comparison between the Laptev interest point detector [18] and our approach.

Dataset	Laptev detector [18] (%)	Our detector (%)
KTH	92.33	**94.33**
WEIZMANN	84.44	**96.66**

comparative study [36], it has been shown that this detector outperforms a number of alternatives including the Dollar et al. detector. In our experiment, we also compared this detector with our detector using the same action representation (COP and BOW fusion) over both KTH and WEIZMANN datasets. The original implementation for 3D Harris interest point detector was employed,[5] with the default parameter setting. Table 6 shows that our detector outperforms the Laptev detector. The Laptev detector particularly struggled with the WEIZMANN dataset which has larger number of action classes and less training samples. It is noted that without frame differencing, the Laptev detector is indeed more effective for KTH dataset compared to the Dollar detector. However, although it is more robust against camera movements, it still suffers from the problems of being unable to capture slow movements, and sensitive to highly textured areas in background.

Effects of feature selection—Our MKL fusion method was evaluated in three scenarios: without feature selection, with the proposed feature selection approach (Section 4.3), and with a more complex minimal-redundancy-maximal-relevance (mRMR) algorithm proposed in [28]. Table 7 shows that feature selection improves the recognition performance and the best performance is obtained when the proposed feature selection method is employed. Note that a major attraction of the mRMR method, as compared with other existing feature selection methods, is its low computational cost. Our feature selection has an even lower computational cost. Specifically, our method took less than one twelfth of the time used by the mRMR method for selecting the same amount of features (7.1 s using our method to measure and rank 2500 features, as compared with 90 s using mRMR on a 2.1G PC platform with 4G RAM).

Table 7
Performance comparison between different feature selection approaches.

Dataset	No feature selection (%)	mRMR [28] (%)	Our method (%)
KTH	93.00	92.83	**94.34**
WEIZMANN	93.33	94.44	**96.66**

Each selected feature belongs to either one of the eight scale dependent feature types, defined in Eq. (9), or the two scale-independent features (O_t^r and O_t^{Sp}). Fig. 9 indicates which types of features are more informative than others, according to how many percent of the final selected features they account for. The result seems to suggest that all features are useful. There are, however, some features that are selected relatively more frequently than the other features for both datasets. For instance, O_t^{Sp} measures the absolute speed of the actor in the image frame. In both the KTH and WEIZMANN dataset, the action classes fall in two broad categories: actions involving global body movements (e.g. walking, galloping, skipping) and actions involving localised movements (e.g. waving, jumping bending). For an action in the former category, the actor moves from one side of the image frame to another, whilst in an action from the latter category, the actor remains near the centre of the frame. As a result, global actions such as walking will have a long 'tail' of interest point cloud with a low density. In contrast, localised actions such as hand-clapping will have a denser cloud centred near the actor (see Fig. 4). Therefore, actions belonging to the two different categories will have very different O_t^{Sp} values because for the localised movement actions, O_t^{Sp} will be very close to zero. This means that O_t^{Sp} will be a good feature for separating these two categories of actions and should be selected more for action recognition. Similarly, C_s^D (cloud density) and C_s^{Vd} (vertical distance between target and cloud) are also good features for separating these two categories of actions, and a relatively high percentage of the final selected features are from them as shown in Fig. 9.

Evaluation of the outliers filtering parameter—As described in Section 4.2, to remove erroneous interest points from the background area, an outlier filtering step is employed before the feature extraction phase. Specifically, if the distance between an interest point and the cloud centroid is greater than l times the average distance over all points, it is deemed as an outlier and removed subsequently. In the experiments reported so far, we set the value of l to be 4. In Fig. 10 we investigate how different values of l will affect the recognition performance. It can be seen that the recognition performance is in general insensitive to the parameter l.

Effect of the codebook size for BOW representation—As mentioned in Section 7.2, the Bag of Words (BOW) features are extracted by computing a 3D Gradient descriptor for each interest point and constructing a codebook using k-means clustering. The codebook sizes used in different existing works vary drastically. In this work, we set the size of codebook to be 300 for KTH and 250 for WEIZMANN for the BOW representation. This is based on the suggestion by Dollar et al. [9]. They empirically found out that a codebook size of between 200 and 500 is suitable for action representation. We have carried out a similar evaluation on the effect of different codebook size on the recognition performance. As can be seen in Fig. 11, our finding is similar to that in [9]. That is, the optimal size is most likely to be within the range of 200–500.

Effects of less training data—In order to investigate the performance of our approach given less training data, in this experiment,

[5] www.irisa.fr/vista/Equipe/People/Laptev/download.html

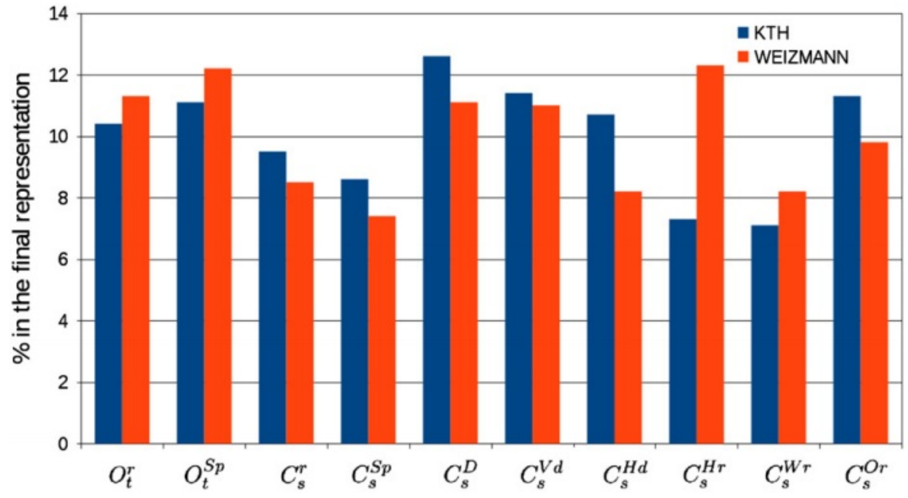


Fig. 9. The percentage of each type of features selected for the final COP representation.

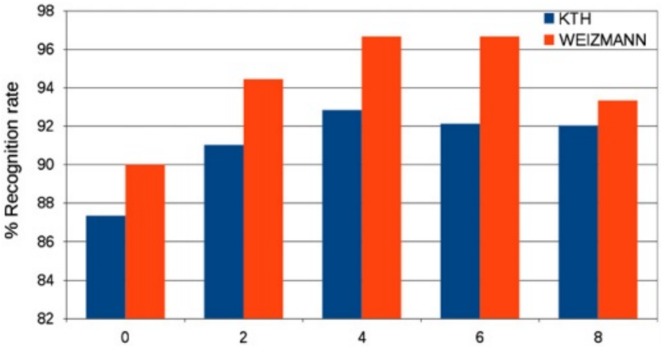


Fig. 10. Effect of the interest point outliers detection threshold on the action recognition performance.

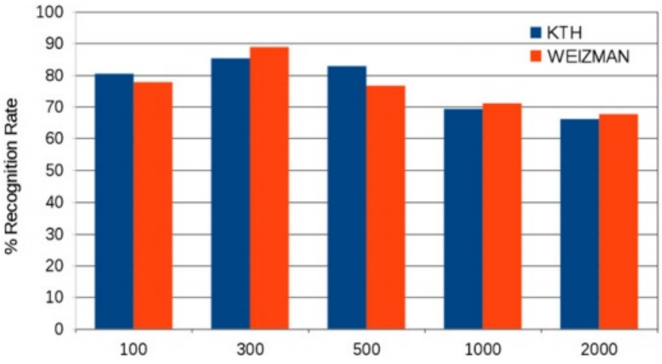
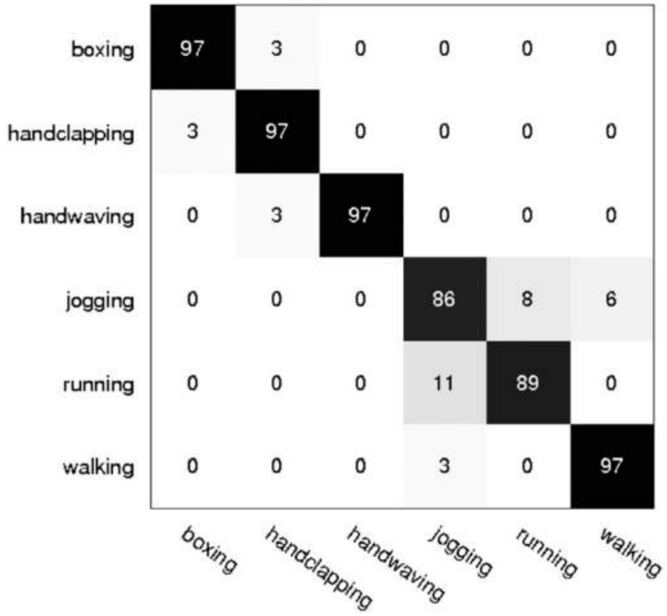


Fig. 11. Influence of the BOW codebook size on the recognition performance.



	boxing	handclapping	handwaving	jogging	running	walking
boxing	97	3	0	0	0	0
handclapping	3	97	0	0	0	0
handwaving	0	3	97	0	0	0
jogging	0	0	0	86	8	6
running	0	0	0	11	89	0
walking	0	0	0	3	0	97

Fig. 12. Confusion matrix on the KTH dataset with less training data (16 subjects for training and nine for testing).

the training set consists of action sequences of 16 subjects instead of 24 for the default Leave-One-Out setting. An average recognition rate of 93.98% is obtained for the KTH dataset and the confusion matrix is presented in Fig. 12. As expected, with less training data, the performance is worse (compared with 94.33% obtained using 25 subjects for training). However, the decrease in performance is very small. In contrast, when we use the Laptev detector, the performance decreases from 92.33% to 87.50% when less training data is used, which is a much larger decrease. This suggests that our detector is less sensitive to the small training data size.

Processing time—During training, most of the computation time was spent on feature extraction. Specifically for each leave-one-out run, on average the amount of time required for feature extraction was 1106.30 s for WEIZMANN and 37 399.10 s for KTH (there are much more training clips in KTH than WEIZMANN). After feature extraction and selection, the training of the multiple kernel SVM classifier was much faster, needing on average 0.34 s for WEIZMANN and 3.34 s for KTH. During testing, the average processing times for each test clip on the WEIZMANN dataset were: 12.80 s on feature extraction and 0.0017 s for classification. For the KTH dataset those numbers became 64.92 and 0.0019 respectively. All implementations were in Matlab on a 2.1G PC platform with 4G RAM.

7.4. Robustness evaluation

The robustness of our method was evaluated using the WEIZMANN robustness test sequences. Examples of the detected clouds of interest points are shown in Fig. 13. The result is reported in Table 8. It can be seen that the BOW based representation is very sensitive to view angle, variations in action, and occlusions, with only half of the test sequences being recognised

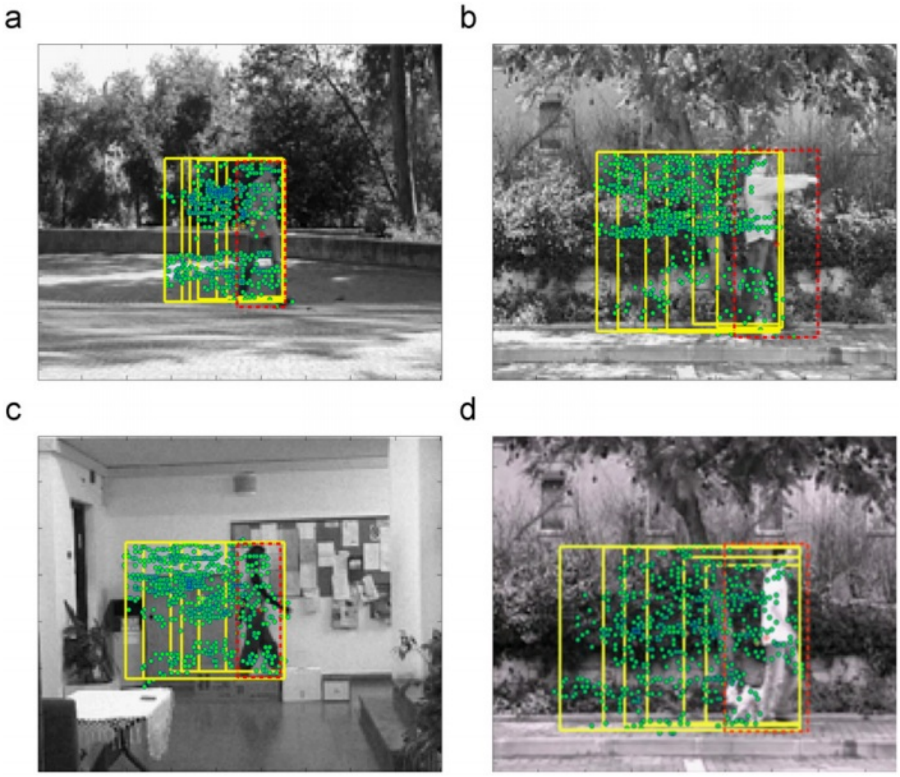


Fig. 13. Example of Clouds of Points detected in the sequences used in the robustness test experiments. (a) Walking along 45°, (b) sleepwalking, (c) walking with occluded legs, (d) walking with a dog.

Table 8
Robustness test result.

Dataset	Correct recognition
COP+BOW	**20 out of 20**
COP	19 out of 20
BOW	10 out of 20
Blank et al. [4]	19 out of 20
Wang et al. [37]	18 out of 20

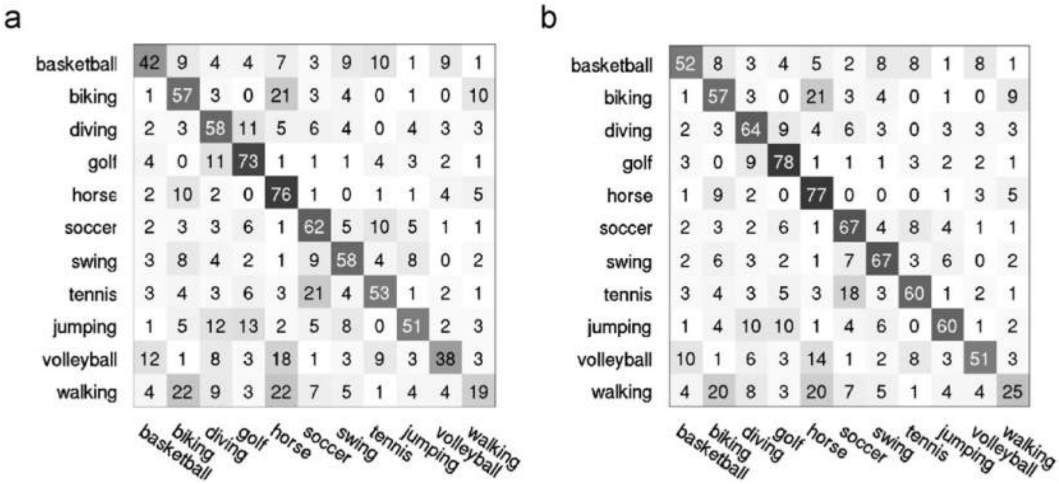
correctly. In contrast, the proposed COP representation is much more robust, with only a single misclassified sequence (a person walking with a dog was recognised as skipping). In the sequence, the most informative human body part for the action (i.e. the legs) overlapped with another object (the dog), which was also walking but in a very different way (see Fig. 13(d)). With COP and BOW feature fusion, all the sequences are correctly classified. This result suggests that BOW and COP feature fusion applied in real-world complex scenarios improves the robustness of action recognition. In particular, by merging two complementary representations such as COP and BOW, it is possible to overcome partial occlusions and action distortions caused by other objects (for example, those caused by the dog in the walking-with-the-dog sequence). Table 8 shows that our method also outperforms existing action recognition approaches that have reported results on this robustness test dataset.

7.5. Addressing more challenging action recognition problems

As the final experiment, we tested the proposed framework on a more challenging dataset, namely the YouTube "action in the wild" dataset [24]. It is the most extensive realistic action dataset available to public. The dataset is composed of 1168 videos collected from YouTube. These videos contain a representative collection of real world challenges such as shaky cameras, cluttered background, variation in object scale, variable and changing view-point and illumination, and low resolution. Particularly, since these videos are mostly captured by hand-held cameras, the camera movements are much more drastic and unpredictable compared the other two datasets. The YouTube dataset contains 11 action categories: basketball shooting, volleyball spiking, trampoline jumping, soccer juggling, horse-back riding, cycling, diving, swinging, golf, swinging, tennis swinging, and walking. Clips have different frame rates but a fixed frame size of 320 by 240 pixels. The clips last between 3 and 15 s.

We obtained an average recognition accuracy of 55.04% using the proposed method. The confusion matrix is shown in Fig. 14(a). This performance is inferior to that best result reported in [24] (71.2%). This is not surprising because our method is not designed for coping with the drastic camera motions and dynamic background featured in the YouTube dataset. In particular, it is observed that when the camera movements and moving background objects are less an issue (e.g. horse-back riding, soccer juggling and swinging), our method outperforms that in [24].

There are various possible techniques that can be considered to make our method more suitable to a more challenging action recognition task. In particular, as suggested in [24], the following steps have proven to be effective in dealing with camera motions and dynamic background: (1) detecting dense 2D static appearance features, and fusing with space-time interest point based features, (2) performing feature pruning and feature mining/ranking, (3) semantic visual vocabulary learning, and (4) detecting a region of interest. All four steps could be integrated with the proposed method to improve the performance. To demonstrate this, we have implemented a simple method on region of interest (ROI) detection as proposed in [6]. Specifically, in each frame the ROI centroid $(\hat{x},\hat{y})$ is computed by averaging spatial coordinates of all interest points detected; then the ROI dimensions are given by $D_x = 2\sqrt{2c_{xx}}$ and



a

	basketball	biking	diving	golf	horse	soccer	swing	tennis	jumping	volleyball	walking
basketball	42	9	4	4	7	3	9	10	1	9	1
biking	1	57	3	0	21	3	4	0	1	0	10
diving	2	3	58	11	5	6	4	0	4	3	3
golf	4	0	11	73	1	1	1	4	3	2	1
horse	2	10	2	0	76	1	0	1	1	4	5
soccer	2	3	3	6	1	62	5	10	5	1	1
swing	3	8	4	2	1	9	58	4	8	0	2
tennis	3	4	3	6	3	21	4	53	1	2	1
jumping	1	5	12	13	2	5	8	0	51	2	3
volleyball	12	1	8	3	18	1	3	9	3	38	3
walking	4	22	9	3	22	7	5	1	4	4	19

b

	basketball	biking	diving	golf	horse	soccer	swing	tennis	jumping	volleyball	walking
basketball	52	8	3	4	5	2	8	8	1	8	1
biking	1	57	3	0	21	3	4	0	1	0	9
diving	2	3	64	9	4	6	3	0	3	3	3
golf	3	0	9	78	1	1	1	3	2	2	1
horse	1	9	2	0	77	0	0	0	1	3	5
soccer	2	3	2	6	1	67	4	8	4	1	1
swing	2	6	3	2	1	7	67	3	6	0	2
tennis	3	4	3	5	3	18	3	60	1	2	1
jumping	1	4	10	10	1	4	6	0	60	1	2
volleyball	10	1	6	3	14	1	2	8	3	51	3
walking	4	20	8	3	20	7	5	1	4	4	25

Fig. 14. Confusion matrix obtained on the YouTube dataset [24]. (a) An average recognition rate of 55.04% was obtained using the proposed method and (b) with an additional region of interest detection step, the result was improved to 61.07%.

$Dy = 2\sqrt{2c_{yy}}$ where c_{xx} and c_{yy} are the second central moments of the points. With this simple pre-processing step, the average recognition rate was increased to 61.07% (see Fig. 14(b)).

8. Discussions and conclusions

We have proposed a novel action representation method which differs significantly from the existing interest point based representation in that only the global distribution information of interest points is exploited. In particular holistic features from clouds of interest points accumulated over multiple temporal scales are extracted. Since the proposed spatio-temporal distribution representation contains different but complementary information to the conventional Bag of Words (BOW) representation, we formulate a feature fusion method based on multiple kernel learning. Experiments using the KTH and WEIZMANN datasets demonstrate that our approach outperforms most existing methods in particular under occlusion and changes in view angle, clothing, and carrying condition.

Failure mode—For the KTH dataset, the errors made by our approach come mainly from three classes: jogging, running, and walking which are visually very similar. With the global features extracted using our Clouds of Points representation, less errors were made compared to a conventional interest point based method (see Fig. 7). However, there are still misclassifications between jogging and running because there is no clear separation between these two action classes—running slowly becomes jogging and how slow it should be depends on human interpretation thus is subjective. As for the WEIZMANN dataset, our method tends to mistaken skipping as jumping or running, and side walking as walking. Again skipping is a combination of jumping and running, and side walking is visually very similar to walking. In order to avoid these mistakes, one could build a human body model and separate different body parts in the representation. Alternatively, one could extract features from 3D human body shapes. However, as we stated in Section 2, both model tracking based approaches and spatio-temporal shape template based approaches require highly detailed silhouettes to be extracted. They thus stand no chance on noisy data such as the KTH dataset.

Action classification vs action detection—Similar to most existing action recognition work, the approach described in this paper is designed to address the action classification problem. In particular, it is assumed that each video clip contains a single action belonging to one of the known categories. In other words, it is assumed that temporal segmentation is done and there are no multiple different actions co-existing in the clip. For these methods, the decision is made for the whole clip and only after the whole clip is observed. Note that some of these approaches can also be applied for online detection (e.g. [26]). More recently researchers start to look at the problem of action detection, that is to recognise and localise actions both spatially and temporally with other actions performed in the background [39,16,42]. They in general adopt a 3D sliding window approach similar to the 2D sliding window approach used for static object detection in images. The proposed method could be extended to address the action detection problem by adopting a 3D sliding window strategy.

Future work—There are a number of areas that require further investigation. First, the current work only considers fusion of features extracted from interest points. Recently it has been demonstrated that alternative features such as Bag of Optical Flows (BOF), feature trajectory context, and higher-level contextual features such as object detector scores and scene context give strong performance on benchmarking datasets [24,34]. These features could contain information that is highly complementary to interest point based features and thus should be combined together. Second, although our feature selection method is effective in removing irrelevant features and improving recognition performance, it selects a single set of features for all action classes. Nevertheless, the optimal set of features for distinguishing different set of action classes could be different. Therefore a more fine-grained feature selection method needs to be investigated. Finally, as we mentioned earlier, our approach is unable to cope with actions performed by multiple objects simultaneously and in front of a dynamic crowded background since it is an action classification method. Ongoing work includes developing an action detector based on the Clouds of Points representation proposed in this paper.

Acknowledgements

This work was supported by the UK Engineering and Physical Sciences Research Council (grant number EP/E028594/1).

References

[1] A. Ali, J. Aggarwal, Segmentation and recognition of continuous human activity, in: IEEE Workshop on Detection and Recognition of Events in Video, 2001, p. 28.

[2] F. Bach, G. Lanckriet, M. Jordan, Multiple kernel learning, conic duality, and the SMO algorithm, in: ICML, 2004.

[3] C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[4] M. Blank, L. Gorelick, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, ICCV, vol. 2, 2005, pp. 1395–1402.

[5] M. Bregonzio, S. Gong, T. Xiang, Action recognition with cascaded feature selection and classification, in: ICDP, 2009.

[6] M. Bregonzio, J. Li, S. Gong, T. Xiang, Discriminative topics modelling for action feature selection and recognition, in: BMVC, 2010.

[7] C. Schüldt, I. Laptev, B. Caputo, Recognizing human actions: a local SVM approach, ICPR, vol. 3, 2004, pp. 32–36.

[8] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: CVPR, 2005.

[9] P. Dollar, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: VS-PETS, 2005.

[10] A. Efros, A. Berg, G. Mori, J. Malik, Recognizing action at a distance, in: ICCV, 2003, pp. 726–733.

[11] A. Fathi, G. Mori, Action recognition by learning mid-level motion features, in: CVPR, 2008.

[12] P. Felzenszwalb, D. McAllester, D. Ramanan, A discriminatively trained, in: CVPR, 2008.

[13] P. Gehler, S. Nowozin, On feature combination for multiclass object classification, in: ICCV, 2009.

[14] A. Gilbert, J. Illingworth, R. Bowden, Scale invariant action recognition using compound features mined from dense spatio-temporal corners, ECCV, vol. 1, 2008, pp. 222–233.

[15] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, IEEE Transaction on Pattern Analysis and Machine Intelligence 29 (12) (2007) 2247–2253.

[16] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, T.S. Huang, Action detection in complex scenes with spatial and temporal ambiguities, in: ICCV, 2009.

[17] N. Ikizler, R. Gokberk Cinbis, P. Duygulu, Human action recognition with line and flow histograms, in: ICPR, 2008.

[18] I. Laptev, On space-time interest points, in: IJCV, 2005.

[19] Y. Ke, R. Sukthankar, M. Hebert, Efficient visual event detection using volumetric features, ICCV, vol. 1, 2005, pp. 166–173.

[20] A. Kläser, M. Marszałek, C. Schmid, A spatio-temporal descriptor based on 3D-gradients, in: BMVC, 2008.

[21] P. Kontkanen, P. Myllymaki, MDL histogram density estimation, in: Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, 2007.

[22] Z. Lin, Z. Jiang, L.S. Davis, Recognizing actions by shape-motion prototype trees, in: ICCV, 2009.

[23] J. Liu, M. Shah, Learning human actions via information maximization, in: CVPR, 2008.

[24] J. Liu, J. Luo, M. Shah, Recognizing realistic actions from videos "in the wild", in: CVPR, 2009.

[25] J. Liu, Y. Yang, M. Shah, Learning semantic visual vocabularies using diffusion distance, in: CVPR, 2009.

[26] J. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human action categories using spatial-temporal words, International Journal of Computer Vision 79 (3) (2008) 299–318.

[27] J. Parker, Algorithms for Image Processing and Computer Vision, Wiley Computer Publishing, 1997.

[28] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transaction on Pattern Analysis and Machine Intelligence 2 (2005) 1226–1238.

[29] D. Ramanan, D.A. Forsyth, Automatic annotation of everyday movements, in: NIPS, 2003.

[30] C. Rao, M. Shah, View-invariance in action recognition, CVPR, vol. 2, 2001, pp. 316–322.

[31] S. Savarese, A. Del Pozo, J. Niebles, L. Fei-Fei, Spatial-temporal correlations for unsupervised action classification, in: IEEE Workshop on Motion and Video Computing, 2008.

[32] Y. Sheikh, M. Sheikh, M. Shah, Exploring the space of a human action, in: ICCV, 2005.

[33] S. Sonnenburg, G. Rätsch, C. Schäfer, B. Schölkopf, Large scale multiple kernel learning, in: JMLR, 2006.

[34] J. Sun, X. Wu, S. Yan, L. Cheong, T. Chua, J. Li, Hierarchical spatio-temporal context modeling for action recognition, in: CVPR, 2009.

[35] X. Sun, M. Chen, A. Hauptmann, Action recognition via local descriptors and holistic features, in: CVPR, 2009.

[36] H. Wang, M.M. Ullah, A. Kläser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, in: BMVC, 2009.

[37] L. Wang, D. Suter, Learning and matching of dynamic shape manifolds for human action recognition, IEEE Transactions on Image Processing 16 (6) (2007) 1646–1661.

[38] Y. Wang, G. Mori, Max-margin hidden conditional random fields for human action recognition, in: CVPR, 2009.

[39] W. Yang, Y. Wang, G. Mori, Efficient human action detection using a transferable distance function, in: Asian Conference on Computer Vision, 2009.

[40] B. Yao, S. Zhu, Learning deformable action templates from cluttered videos, in: ICCV, 2009.

[41] A. Yilmaz, M. Shah, Actions sketch: a novel action representation, in: CVPR, 2005, pp. 984–989.

[42] J.S. Yuan, Z.C. Liu, Y. Wu, Discriminative subvolume search for efficient action detection, in: CVPR, 2009, pp. 2442–2449.

[43] Z. Zhang, Y. Hu, S. Chan, L. Chia, Motion context: a new representation for human action recognition, ECCV, vol. 4, 2008, pp. 817–829.

[44] Z. Zhao, A. Elgammal, Information theoretic key frame selection for action recognition, in: BMVC, 2008.

Matteo Bregonzio received the Master degree in Telecommunication Engineering from Politecnico di Milano in 2006. He is now a Ph.D candidate (Supervisors: Shaogang Gong and Tao Xiang) as part of the BEWARE project in the Department of Computer Science, Queen Mary, University of London. His current research interests include video processing, machine learning and human action recognition.

Tao Xiang received the Ph.D. degree in electrical and computer engineering from the National University of Singapore in 2002. He is a currently a lecturer in the Department of Computer Science, Queen Mary, University of London. His research interests include computer vision, statistical learning, video processing, and machine learning, with focus on interpreting and understanding human behaviour.

Shaogang Gong is Professor of Visual Computation at Queen Mary, University of London, a Fellow of the Institution of Electrical Engineers and a Member of the UK Computing Research Committee. He received his DPhil in computer vision from Keble College, Oxford University in 1989. He has published over 200 papers in computer vision and machine learning, and a book on Dynamic Vision: From Images to Face Recognition. His work focuses on motion and video analysis; object detection, tracking and recognition; face and expression recognition; gesture and action recognition; visual behaviour profiling and recognition.

Please cite this article as: M. Bregonzio, et al., Fusing appearance and distribution information of interest points for action recognition, Pattern Recognition (2011), doi:10.1016/j.patcog.2011.08.014