Journal of Electronic Imaging

JElectronicImaging.org

Real-time action recognition using a multilayer descriptor with variable size

Marlon F. Alcantara
Thierry P. Moreira
Helio Pedrini

Journal of Electronic Imaging 25(1), 013020 (Jan/Feb 2016)

Real-time action recognition using a multilayer descriptor with variable size

Marlon F. Alcantara, Thierry P. Moreira, and Helio Pedrini*
University of Campinas, Institute of Computing, Av. Albert Einstein 1251, Campinas-SP 13083-852, Brazil

Abstract. Video analysis technology has become less expensive and more powerful in terms of storage resources and resolution capacity, promoting progress in a wide range of applications. Video-based human action detection has been used for several tasks in surveillance environments, such as forensic investigation, patient monitoring, medical training, accident prevention, and traffic monitoring, among others. We present a method for action identification based on adaptive training of a multilayer descriptor applied to a single classifier. Cumulative motion shapes (CMSs) are extracted according to the number of frames present in the video. Each CMS is employed as a self-sufficient layer in the training stage but belongs to the same descriptor. A robust classification is achieved through individual responses of classifiers for each layer, and the dominant result is used as a final outcome. Experiments are conducted on five public datasets (Weizmann, KTH, MuHAVi, IXMAS, and URADL) to demonstrate the effectiveness of the method in terms of accuracy in real time. *©2016 SPIE and IS&T* [DOI: 10.1117/1.JEI.25.1.013020]

Keywords: adaptive learning; action detection; motion silhouettes; surveillance systems; real-time video analysis.
Paper 15533 received Jul. 2, 2015; accepted for publication Jan. 12, 2016; published online Feb. 3, 2016.

1 Introduction

Surveillance systems have a wide range of applications and can be used in tasks such as crime prevention, accident monitoring, personal identification, and vandalism prevention, among several others.[1] Through the images obtained by video cameras processed by a monitoring system, it is possible to control activities in complex scenarios and with a large concentration of people, which could be impracticable for a human operator.

The development of digital technology has promoted substantial progress in the area of visual surveillance. Cameras have been developed at higher resolutions, smaller dimensions, and higher frame rates. Videos acquired by cameras have been recorded in larger quantities due to the increase in storage capacity of the digital media.

In general, current research[2–10] focuses on the development of intelligent surveillance systems that aim at interpreting human activity instead of using a passive monitoring system, which is the most commonly employed technology. Intelligent systems may allow the reduction of the necessity of monitoring operators and can help in the analysis of images and videos. Nevertheless, intelligent monitoring systems should be capable of automatically extracting complex information of the observed scene and classifying its main events.

The term "actions" is not clearly defined in the literature. However, in general, "action" means a simple pattern of human movement, such as walking or taking steps, waving hands, or collapsing, and can afterward be used to infer a complex task that involves the identification of several actions, interaction between individuals, and interactions with objects in the scene.[11–13] Despite the various definitions for actions, most methods available in the literature commonly use video scenes containing only one action per frame.

The identification of human actions refers directly to the comprehension of human behavior. This understanding involves modeling and classifying actions within a restricted set of rules. The main strategy for this problem is to divide human actions into stages and classify them. The automatic analysis and classification of actions from surveillance cameras can aid or, sometimes, substitute for the human monitoring operator. An effective monitoring system can promote the replacement of current passive systems employed in surveillance and improve the identification of events of interest.

This paper describes a real-time action method based on adaptive training of a new multilayer descriptor with variable size that is applied multiple times to a single classifier. The algorithm assumes that a form of cumulative motion shapes (CMSs)[14] can provide enough information about the action being performed in a video stream. To deal with different possible scenarios of an action occurrence, a set of CMSs is extracted according to the number of frames present in the video. Each CMS is used as an individual entity in the training stage, whose descriptor can have a variable size.

The major contributions of the current work compared against the approach developed by Alcantara et al.[15] include: (i) the proposition of a multilayer descriptor (the descriptor in Ref. 15 has one layer); (ii) the descriptor's ability to self-adjust to a nonsegmented scene with variable duration and, due to that, the descriptor's applicability to more datasets; (iii) the combination of multiple responses of classifiers and decisions for the most representative outcome as the final verdict. The proposed action identification method is

*Address all correspondence to: Helio Pedrini, E-mail: helio@ic.unicamp.br

evaluated on five public datasets (Weizmann, KTH, MuHAVi, IXMAS, and URADL).

The text is organized as follows. Section 2 describes relevant work related to the topic under investigation. The proposed methodology is explained in Sect. 3. Section 4 presents and discusses the experiments and results obtained with the proposed methodology applied to public benchmarks. Section 5 concludes the paper and includes some directions for future work.

2 Related Work

Several strategies for addressing the action recognition problem have been proposed in the literature, which in this work are classified into three categories: appearance-based, shape-based, and other approaches.

2.1 *Appearance-Based Methods*

Appearance-based methods work by extracting local information around a set of spatio-temporal interest points (STIPs),[16] commonly representing corners in the three-dimensional (3-D) motion volume. Descriptors are usually constructed as cuboids around the STIPs.

Methods such as Laptev's 3-D extension of the Harris operator,[16] Dollár's method,[17] scale-invariant feature transform (SIFT),[18] and speeded-up robust features[19] provide spatio-temporal information. The next step of these approaches is to cluster the descriptors into appearance classes, or vocabularies, and build histograms, usually called bag-of-words (BoW) or bag-of-visual-words (BoVW). The most commonly used clustering algorithm in this process is K-means.[20]

Ryoo and Aggarwal[21] extracted STIPs[17] and clustered them into a dictionary. Two 3-D histograms are assembled with temporal and spatial relationships. The system decides whether the testing video contains an activity or not by measuring the similarities between the video and other training videos containing the activities. Next, for each video of the group, the intersection of the temporal relationship histograms is computed, and each pair of characteristics of the resulting votes for the instants of the beginning and end of the action.

Sun et al.[22] used local and holistic descriptors that are joined before clustering and classification. Local descriptors are two-dimensional (2-D) SIFT and 3-D SIFT, whereas holistic descriptors are Zernike moments in every frame and in motion energy images. Descriptors are concatenated and clustered to create a dictionary.

Ta et al.[23] developed a method that forms pairwise groupings in spatio-temporal information (pairwise features, PWF). The interest points and spatio-temporal features are extracted through the STIP method developed by Dollár et al.[17]

Local appearance information of both STIPs is concatenated to form the PWF appearance descriptor, and a vector from the first point to the second one forms the geometrical descriptor. Clustering and BoW processes are applied to each descriptor, forming two histograms per action sequence that are concatenated into one feature vector used for a support vector machine (SVM) to classify the actions.

Wu et al.[24] developed a hierarchical action recognition framework. The first level recognizes poses, or coarse level actions, such as standing, sitting, and lying, which is performed through the aspect of the bounding box by 3-D estimation. Actions are refined by combining the BoW strategy to the location in which the action occurs. Finally, three strategies (best view, combined view, and mixed view) are employed to allow multiview.

Bregonzio et al.[25] used the global distribution information of interest points to acquire geometrical information of the action, where actions are represented as clouds of interest points accumulated at different temporal scales. Interest points are accumulated over time at different time scales to form multiple clouds. Features are computed from the clouds and fused to an appearance descriptor based on BoW through multiple kernel learning.

Zhang and Tao[26] used slow feature analysis (SFA),[27] where cuboids are extracted from randomly chosen points over movement silhouettes. To improve temporal information on cuboids, they are transformed in a sequence of three frames. In addition to the original unsupervised SFA, three other models are proposed. Accumulated squared derivatives are computed from the outputs to measure the fitting degree from cuboids to the slow feature functions. A linear multiclass SVM is used in the classification.

Onofri and Soda[28] extracted features of video portions with the MoSIFT method[29] and constructed a BoVW. Then, an information bottleneck[30] is applied to reduce dimensionality. A multiple subsequence combination is applied by building a matrix containing the probabilities of each class for each subsequence. Four criteria are applied on the probabilities of each class, and the class with the best output is selected.

Chen et al.[31] used spatio-temporal characteristics, called Lie Algebrized Gaussians, based on Gaussian mixture models. The work also analyzes the actions by midlevel characteristics, where actions are modeled in a 3-D form such that the video frames represent the first 2-D, whereas time represents the third one. Although the work is not innovative in the way an action is represented, in its validation protocol by separating the KTH dataset in distinct environments, the work is one of the most accurate found in the literature.

2.2 *Shape-Based Methods*

The nature of shapes can be very distinctive, for instance, human silhouettes, movement shapes, relative positions of body parts, or pose estimation. Such approaches frequently use movement segmentation to obtain the silhouette or to narrow down other searches. These methods are often fragile to conditions that hinder motion segmentation, such as lighting variations, and are not robust to occlusions. On the other hand, they usually result in simpler, yet meaningful, descriptors, which may allow faster execution.

Singh et al.[32] used silhouettes for action recognition. A minimum size to fit all silhouettes over time is computed, and a new space-time volume is built with the computed size and the time span of the original sequence. The frames are divided into a grid, generating subvolumes. A mean-power spectrum is calculated from the frequency spectrum of each pixel in the bins. All vectors are concatenated to build a final descriptor.

In the method developed by Raja et al.,[33] a small set of frames is manually annotated, leaving the program to annotate the remaining ones. In each frame, the positions of the head, hands, and feet are located with respect to a bounding

box. A description of the pose is made by maximizing an energy function. A graph is constructed by linking each labeled image to its nearest unlabeled neighbors. Then, unlabeled images are linked to their nearest neighbors (NN), labeled or unlabeled. Images are then labeled by optimizing the global energy of the graph.

Hsieh et al.[34] presented a silhouette-based method that represents the shape by histograms. The silhouette is extracted by adaptive background subtraction and mapped into three polar coordinate systems. The first circle includes the whole silhouette, the second only the top part (arms and head), and the third only the bottom part (legs). The polar systems are partitioned into bins and the silhouette histograms are computed by counting the number of pixels in each bin. The histograms are concatenated to build the pose descriptor.

Cheema et al.[35] developed a method that uses weighted key poses to recognize actions in videos. Pose representation is obtained by a normalized distance function over the sampled contour points. Key poses are computed for each action through K-means clustering, and weights are assigned to each one according to its ambiguity by counting its occurrence in other classes. For a sequence with multiple frames, a weighted voting scheme is used, whereas for a single image, simple key pose matching is done.

Karthikeyan et al.[36] described silhouettes by a 2-D Radon transform and their velocity. For each signature, eigen mode and multiset partial least squares mode are computed, resulting in four vectors of 180 dimensions for each camera view, which are concatenated to form the final description. Probabilistic subspace similarity learning was used to perform intraclass and interclass learning.

Guo et al.[37] computed an empirical estimate of the covariance matrix over the features extracted from a video sample. The log-covariance matrix is calculated by reconstructing the matrix using the logarithms of its eigenvalues. Two classification approaches are considered: a nearest-neighbor classification using two Riemannian matrix distance metrics and a sparse linear approximation applied to log-covariance matrices in order to determine the label of the testing sample. Two strategies are adopted to obtain the feature vectors, silhouette tunnel shapes, and optical flow.

Chaaraoui et al.[38] developed a feature subset selection method that separates the relevant parts of the feature vector and excludes subsets that add redundancy to the feature. The descriptor is built by dividing the polar space into radial bins and summarizing the points of each one. The poses are clustered with the K-means algorithm, obtaining the key poses. Each video is represented by the sequences of key poses, and the comparison between videos is performed with dynamic time warping. A genetic algorithm is used to determine which of the bins will be used in the classification.

2.3 *Other Approaches*

Wang et al.[39] implemented a real-time surveillance system robust to horizontal and vertical camera movement. The movement segmentation is carried out with optical flow, the resulting objects are split into a grid, and a histogram of optical flow is calculated for each block by dividing the directions into eight bins. Various statistical values are calculated over each bin, and the descriptor is built by the concatenation of all values. In addition to these features, shape and trajectory are also employed. The system learns the action from every frame, so that each frame of the test videos also has a response. The output for the entire video is generated by a voting scheme.

Junejo and Aghbari[40] developed an approach that uses the trajectory of reference points of actors for action recognition. A method transforms a trajectory time-series into a symbolic representation, such that distances between trajectories are approximated to the distance between their representations. Velocity, acceleration, and curvature information is added to the descriptor to enrich the K-nearest neighbor (K-NN) classifier.

Ji et al.[41] developed a method based on deep learning. Five channels of information are obtained by applying filters that, for each frame, obtain the gray values of the features, x and y directions of gradient and optical flow. Convolutional neural networks and subsampling filters are then applied alternately on a sequence of seven frames; these operations transform the video volume into a feature vector. Global information about the action is also passed to the last neural network layer.

Moghaddam and Piccardi[42,43] contributed to enhancing the action classifier independently of the types of features extracted. The initialization of training parameters of hidden Markov models is crucial to finding optimal parameters in a short processing time.

Focusing on real-time applications, Tran et al.[44] applied a classifier based on cuts in the frequency domain (Fourier transform). The work demonstrates that operations based on shapes require much processing time and have a direct impact on the classifier performance. However, due to this improvement in run-time performance, accuracy was compromised, and it is below other works found in the literature.

Antonucci et al.[45] used imprecise hidden Markov models to classify multivariate time series. Each learned model corresponds to an imprecise mixture of Gaussian densities to reduce the problem to the classification of static information. A K-NN is employed in the classification process. The results obtained for some datasets are not as accurate as other methods found in the literature.

3 Methodology

The main steps of the methodology proposed in this work are shown in Fig. 1 and explained as follows. The algorithm starts with a motion segmentation process based on a background subtraction algorithm, described in Ref. 46. It learns a background model using multiple mixture models for each pixel.

A sliding window is used to build CMSs,[15] which are a simple union of all binary foregrounds of the video frames in a time interval. The CMS for the k'th frame of the video (CMS_k) is given by Eq. (1), where n is the size of the sliding window and S_i is the motion shape obtained from the i'th frame.

$$\mathrm{CMS}_k = \bigcup_{i=k-n}^{k} S_i. \tag{1}$$

The extracted foreground is subject to noise, missegmentation, and disconnections. Therefore, morphological operations are applied to remove such imperfections and discard

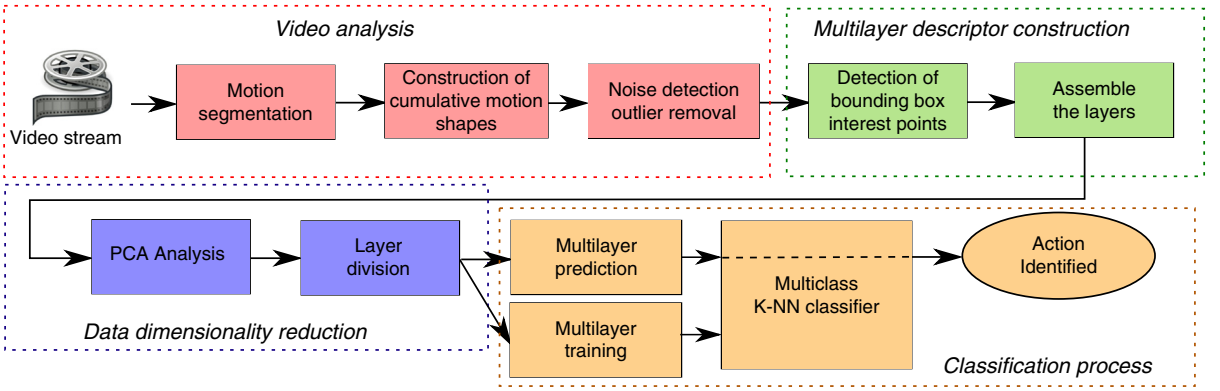

Fig. 1 Diagram illustrating the main steps of the proposed action detection methodology.

certain CMSs. Initially, morphological closing is applied with a 3×3 structuring element to join fragmented shapes. Then, an area opening is used to remove small, noisy objects, usually due to small changes in background and lighting. In the experimental tests, any component with a number of pixels smaller than 0.28% of the image is interpreted as noise or useless information, such that these regions are considered background components. Finally, a morphological reconstruction is used to reattach disconnected parts around the remaining components, recovering some fragments subtracted in the previous step. Some frames with outlier values are also discarded; a frame is considered outlier when the bounding box shows little movement, no movement at all, or when part of the shape is outside the frame. An example of a CMS created after this process is shown in Fig. 2.

Different actions often have common poses. The CMS adds temporal information to poses without increasing the dimensionality, therefore neither requiring more processing power nor memory usage. For missegmentation cases, the CMS sometimes gathers the broken portions of movement, producing a meaningful shape.

To acquire the interest points, the strategy is to select extreme points on CMS, since they are the nearest points from some key points fixed on the bounding box. The number of key points is the same over all video streams used in the training process.

Fig. 2 Example of CMS built through our method. The frames are obtained from MuHAVi.[32]

The interest points for each CMS are obtained as follows. Initially, the bounding box that contains a CMS is found, then the bounding box is subdivided into DX (number of divisions on the x axis) and DY (number of divisions on the y axis), which do not need to be equal for all four sides. Eventually, it could be interesting to use a distinct number for horizontal and vertical sides, since the CMS can have more information disposed in the vertical than the horizontal direction. Each subdivision is called a key point, defined as follows: let c_a, c_b, c_c, and c_d be the four corners of a bounding box in the clockwise direction. The point p in the k'th subdivision between two corners is defined as p_k in Eq. (2), where D means the number of subdivisions in each axis, c refers to a corner, and x and y indicate two adjacent corners in the clockwise direction.

$$p_k = \frac{k(c_x - c_y)}{D} + c_x. \tag{2}$$

The bounding box center is defined as the center of the CMS coordinate system, and all the coordinates are normalized between -1 and 1 following the bounding box limits. The coordinates of each interest point are obtained in the new coordinate system, according to Algorithm 1. Finally, the descriptor vector is built by concatenating all the coordinates. The normalization is necessary since the interest point coordinates depend on the size of the bounding box.

A video sequence usually contains one CMS for each frame. Therefore, several CMSs can be generated from one action sequence, such that this step ends up with multiple descriptors for each input video and multiple layers of features must be passed forward to the classification machine. There is no precedence order among distinct layers from the same video stream. Extracting multiple samples from the same sequence helps in learning actions starting from any part of their periods; e.g., a walking action may start with two feet together or after a step has already been taken.

Comparison of two sets of CMSs with distinct numbers of layers is possible due to an adaptive descriptor developed in this work. The descriptor, shown in Fig. 3, is composed of N CMS, where N is not necessarily the same for all descriptors of the video streams. In contrast to descriptors with fixed dimensions, the use of N CMS gives the descriptor the ability to self-adjust for a nonsegmented scene with variable

Algorithm 1 Function that finds an interest point given a control point.

```
1: function INTEREST POINT (control point)
2:     r←CMS[1]                                   ▷ Interest point candidate
3:     for i = 2 to CMS.size() do
4:       if distance(control point, CMS[i]) < distance(control point, r)
           then
5:         r←CMS[i]                               ▷ Update the candidate
6:       end if
7:     end for
8:     return r                                   ▷ Return interest point
9: end function
```

duration, such that the descriptor is more flexible to be applied in datasets with variable length actions.

Each CMS contains a set of M control points that together represent the CMS silhouettes, where M is the same for all CMSs, such that each control point is used to correlate between distinct descriptors. The descriptor is adaptive since it allows that distinct video streams containing a different number of CMSs can be compared in a classifier through individual training for each CMS descriptor. Algorithm 2 presents pseudocode for the descriptor construction considering the application of the principal component analysis (PCA) technique[47] to reduce the dimensionality of the descriptor.

To operate with this specific type of descriptor, a machine learning technique for fast training is required. We employed the NN since it is commonly applied in real-time systems. Other approaches were evaluated, including SVMs and random forests; however, the entire process executed eight times slower and the results achieved lower accuracy with these techniques.

In the training process, each video populates the same machine learning N times; similarly, when a prediction is made, a set of descriptors sampled from the test sequence is computed. Each one is used for a distinct prediction. The K-NN classifier estimates to which action each descriptor fits better. Each prediction works as a vote, and the one that is dominant corresponds to the final verdict. It is important to mention that the construction of the descriptor is the same for both the training and testing stages.

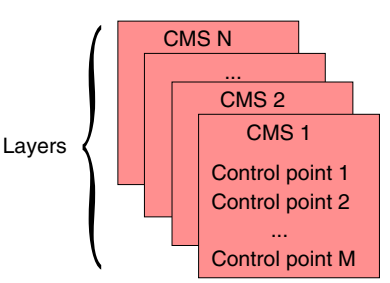

Fig. 3 Structure of the multilayer descriptor.

Algorithm 2 Function that builds the descriptor.

```
1: function BUILD_DESCRIPTOR (CMS_set[...])
2:     descriptor ← {}                            ▷ Initialize descriptor
3:     pca ← PCA(CMS_set)                         ▷ PCA application
4:     for i to pca.size()do                      ▷ All the CMSs are used
5:       descriptor.add(pca[i][1...M])            ▷ Include the first M
                                                    columns
6:     end for
7:     return descriptor                          ▷ Return descriptor
8: end function
```

4 Experimental Results

An i7 computer with 3.5 GHz was used in the experiments, and no parallelism mechanism was implemented. The feature extractor was coded in the C++ programming language with the OpenCV library. The classification code was written separately in the R package through the machine learning libraries e1071 and kernlab. All the experiments were performed using the proposed multitraining with adaptive learning. Five public datasets were used to evaluate our methodology: Weizmann, KTH, MuHAVi, IXMAS, and URADL, which are widely employed as benchmarks in the research field. For all evaluated datasets, a leave-one-out training/prediction was used in our experiments.

Weizmann[48] consists of 10 classes, with nine actors performing each action, sometimes with some actors performing them more than once, resulting in 93 videos. The dataset contains a total of 5701 frames, 228.04 s captured at 25 FPS, with a size of 180×144 pixels. All the actions occur on the same static background.

KTH[49] consists of six classes, with 25 actors performing each action, in four different scenes, with the exception of one person, who performs one action (hand clapping) in only three scenes, resulting in 599 videos. The dataset contains a total of 289,715 frames, 11,375.32 s captured at 25 FPS, with a size of 160×120 pixels. Most videos have camera movement (zooming, panning, and tilting).

MuHAVi[32] (Multicamera Human Action Video Data) consists of 17 classes, with seven actors performing each action, totaling 119 videos. The actions occur in a closed scenario, with eight cameras surrounding it. The dataset contains a total of 134,085 frames, 5368.16 s captured at 25 FPS, with a size of 720×576 pixels. The MuHAVi dataset has a subset of manually annotated sequences (MuHAVi-MAS), in which the frames are binary images of the silhouette locations. It is divided into 14 primitive actions, and it is usually called MuHAVi14 in the literature. This subset, however, has some actions that vary only in direction (for instance, run left and run right) that are rearranged together, forming another subset with eight classes, called MuHAVi8.

IXMAS (INRIA Xmas motion acquisition sequences)[50] contains 13 classes; however, only 11 are used for validation in the literature. The dataset also offers manually annotated silhouettes. The sequences are recorded at a resolution of 390×291 pixels at 23 FPS. The actors freely choose position and orientation to perform the actions, where each action is acquired by five cameras in distinct positions (four side and one top view.)

The URADL (University of Rochester Activities of Daily Living) dataset[51] contains 10 activity daily action classes recorded in high resolution (1280×720 pixels) and 30 FPS. The actions are performed by four distinct actors in an indoor environment with a fixed camera. The dataset offers short sequences containing only the background to be used in a previous learning for segmentation purpose.

Some parameters described in the proposed methodology are adjusted to each dataset: the number of shapes (NS) used to build the CMS, the number of dimensions used as control points in the bounding box (DX and DY), the number of dimensions in the PCA (ND for each CMS), and the values k and k' to be used in the K-NN classifier. The dimensions of the bounding box and the use of PCA reduce the computational cost during the classification process. Value k varies according to the number of samples available in the dataset. The parameters used to achieve the best results are shown in Table 1. The time values shown in the last column correspond to the total time to process all the frames of a video clip.

Table 2 shows the computational time required for the feature extraction and classification stages achieved by our method and the number of frames, as well as the number of frames per second for each dataset. Assuming that the required rate for a surveillance application to operate in real time is 24 frames/s, it is possible to observe that our method is very fast for the majority of the datasets tested in our experiments. The URADL dataset was processed at 10.24 frames/s; however, it is worth mentioning that the resolution of this video sequence is high (1280×720 pixels).

Table 3 shows results for the KTH and Weizmann datasets, Table 4 presents results for the MuHAVi dataset and its variations, Table 5 provides results for the IXMAS dataset, and Table 6 shows results for the URADL dataset.

Table 1 Main parameters employed in our experiments.

Dataset	NS	DX	DY	ND	k	Precision (%)
Weizmann	2	8	4	34	2	98.9
KTH	4	8	8	18	6	91.3
MuHAVi	6	20	10	55	8	91.6
MuHAVi14	4	16	8	35	4	95.6
MuHAVi8	2	16	8	32	2	100.0
IXMAS	50	8	8	30	1	81.1
URADL	60	16	16	25	2	88.0

Table 2 Performance measures for feature extraction and classification stages.

Dataset	Extraction (s)	Classification (s)	Frames	Frames/s
Weizmann	4.85	0.270	5701	1113.48
KTH	1347.38	5.382	289715	214.17
MuHAVi	2850.29	1.504	137085	48.01
IXMAS	865.46	2.308	34155	39.36
URADL	7100,60	4.732	72729	10.24

Table 3 Correct prediction rates (in percentage) for KTH and Weizmann datasets.

	Data Set	
Method	KTH	Weizmann
Ryoo and Aggarwal[21]	93.8	—
Sun et al.[22]	94.0	97.8
Wang et al.[39]	—	93.3
Ta et al.[23]	93.0	94.5
Raja et al.[33]	86.6	—
Hsieh et al.[34]	—	98.3
Cheema et al.[35]	—	91.6
Bregonzio et al.[25]	94.3	96.7
Junejo and Aghbari[40]	—	88.6
Zhang and Tao[26]	93.5	93.9
Onofri and Soda[28]	97.0	—
Chaaraoui et al.[38]	—	90.3
Ji et al.[41]	90.2	—
Guo et al.[37]	98.5	100.0
Moghaddam and Piccardi[43]	—	96.8
Alcantara et al.[14]	—	94.6
Alcantara et al.[15]	90.1	96.8
Tran et al.[44] [a]	87.1	—
Antonucci et al.[45]	72.5	74.7
Chen et al.[31] [a]	97.1	—
Our method	91.3	98.9

[a]It was validated in the KTH dataset using a manual split between the distinct scenarios.

Table 4 Correct prediction rates (in percentages) for MuHAVi and its manually annotated sub-datasets, MuHAVi14 and MuHAVi8.

Method	Dataset		
	MuHAVi	MuHAVi8	MuHAVi14
Wu et al.[24]	69.2[a]	—	—
Singh et al.[32]	—	82.4	97.8
Moghaddam and Piccardi[42]	80.4	—	—
Karthikeyan et al.[36]	88.2	—	—
Cheema et al.[35]	—	95.6	86.0
Moghaddam and Piccardi[43]	92.0	—	—
Chaaraoui et al.[38]	—	97.1	91.2
Chaaraoui and Flórez-Revuelta[52]	—	100.0	98.5
Alcantara et al.[14]	—	94.6	—
Alcantara el al.[15]	89.1	100.0	94.1
Our method	91.6	100.0	95.6

[a]Experiments conducted by Karthikeyan et al.[36]

The Weizmann dataset contains short video sequences and a few videos for training. In this case, our method maximizes the training, providing an effective classification. The MuHAVi dataset presents a larger number of actions; each streaming contains a long video sequence with an action that is not segmented. Nevertheless, it was possible to achieve impressive results through correct discarding and automatic sampling.

Table 5 Correct prediction rates (in percentage) for IXMAS dataset.[50]

Method	Accuracy (%)
Yan et al.[53]	82.5
Farhadi and Tabrizi[54]	58.1
Li and Zickler[55]	81.2
Liu et al.[56]	75.3
Li, Camps and Sznaier[57]	90.5
Junejo et al.[58]	72.7
Weinland, Boyer and Ronfard[59]	57.9
Evgeniou and Pontil[60]	78.2
Huang, Yeh and Wang[61]	57.3
Reddy, Liu and Shah[62]	72.6
Wu and Jia[63]	88.8
Our method	81.1

Table 6 Correct prediction rates (in percentage) for URADL dataset.

Method	Accuracy (%)
Temporal templates[64] [a]	33.0
Spatio-temporal cuboids[17] [a]	36.0
Space-time interest points[65] [a]	59.0
Velocity histories[51]	63.0
Latent velocity histories[51]	67.0
Augmented velocity histories[51]	89.0
Our method	88.0

[a]Experiments conducted by Messing et al.[51]

The KTH dataset is challenging to our algorithm since motion-based methods usually present difficulties in video streaming that contains abrupt changes in light conditions, distinct spot lights, and, mainly, fast camera movements. Despite such facts, our algorithm achieved a competitive accuracy among state-of-the-art methods.

The IXMAS dataset has another scheme to video streams, where all the actions occur in the same video stream. The classifier needs to divide the entire video in many short sequences. The variable size descriptor benefits this process, and all the fragments are trained equally. The final obtained result is among the best results of the literature.

The URADL dataset contains some actions that are difficult to be separated, for instance, answer and dial phone, or eat banana and eat snacks. The tests using leave-one-out may be influenced by the same actor that performs another action. Nevertheless, the results obtained with our method are comparable to the best results available in the literature.

Unlike the method developed by Alcantara et al.,[15] our strategy for creating the CMS descriptor uses all the available information and, due to that, requires more computational time for the processing. However, our results (in terms of accuracy) are superior compared to those in Ref. 15 for the majority of the tested datasets.

Each tested dataset has its singular characteristics, such as action type, number of actions, and sequence length. The results with multitraining using adaptive learning have been demonstrated to be very effective in terms of accuracy, robustness, and flexibility.

5 Conclusions

Video-based action identification is a challenging problem, such that the development of a robust algorithm that fits well to any possible action and environment is a complex task. Multitraining provides the possibility of partitioning the classification process, where multiple requests to the K-NN classifier can avoid certain false clues and allow a more effective decision. This strategy allowed our method to have competitive results in terms of accuracy compared to the state-of-the-art approaches.

The CMS and the bounding box descriptor[15] provided a complete representation of a motion sequence that could be used for a multitraining purpose where individual motion silhouettes cannot offer sufficient information to independently learn a video action. The proposed method is composed of disjoint modules, such that it is possible to apply specific parts of the method to other descriptors, classifiers, and training processes.

The adaptive learning with multilayer descriptors was demonstrated to work in real time with good accuracy in short and long video streams, such as Weizmann and MuHAVi, respectively. Furthermore, the descriptor applied to the multitraining classifier provides satisfactory results in datasets containing few actions, such as KTH, and containing many actions, such as MuHAVi.

Acknowledgments

The authors are grateful to FAPESP and CNPq for the financial support.

References

1. W. Hu et al., "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Syst. Man Cybern. C* **34**, 334–352 (2004).
2. S. Theusner, M. de Lussanet, and M. Lappe, "Action recognition by motion detection in posture space," *J. Neurosci.* **34**(3), 909–921 (2014).
3. J. Wu, D. Hu, and F. Chen, "Action recognition by hidden temporal models," *Vis. Comput.* **30**(12), 1395–1404 (2014).
4. X. Yang and Y. Tian, "Effective 3D action recognition using eigenjoints," *J. Visual Commun. Image Represent.* **25**(1), 2–11 (2014).
5. A. P. Twinanda et al., "Data-driven spatio-temporal RGBD feature encoding for action recognition in operating rooms," *Int. J. Comput. Assisted Radiol. Surg.* **10**(6), 737–747 (2015).
6. B. Fernando et al., "Modeling video evolution for action recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition* (2015).
7. M. Hoai and A. Zisserman, "Improving human action recognition using score distribution and ranking," *Lect. Notes Comput. Sci.* **9007**, 3–20 (2015).
8. W. Liu et al., "Multiview hessian regularized logistic regression for action recognition," *Signal Process.* **110**, 101–107 (2015).
9. T. V. Nguyen, Z. Song, and S. Yan, "STAP: spatial-temporal attention-aware pooling for action recognition," *IEEE Trans. Circuits Syst. Video Technol.* **25**(1), 77–86 (2015).
10. Y. Song, S. Liu, and J. Tang, "Describing trajectory of surface patch for human action recognition on RGB and depth videos," *IEEE Signal Process Lett.* **22**(4), 426–429 (2015).
11. P. Turaga et al., "Machine recognition of human activities: a survey," *IEEE Trans. Circuits Syst. Video Technol.* **18**(11), 1473–1488 (2008).
12. S. Theusner, M. de Lussanet, and M. Lappe, "Action recognition by motion detection in posture space," *J. Neurosci.* **34**(3), 909–921 (2014).
13. X. Yang and Y. Tian, "Effective 3D action recognition using eigenjoints," *J. Visual Commun. Image Represent.* **25**(1), 2–11 (2014).
14. M. F. Alcantara, T. P. Moreira, and H. Pedrini, "Motion silhouette-based real time action recognition," in *18th Iberoamerican Congress on Pattern Recognition*, Vol. 8259, pp. 471–478, Havana, Cuba (2013).
15. M. F. Alcantara, T. P. Moreira, and H. Pedrini, "Real time action recognition based on cumulative motion shapes," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 2917–2921, Florence, Italy (2014).
16. I. Laptev, "On space-time interest points," *Int. J. Comput. Vision* **64**(2–3), 107–123 (2005).
17. P. Dollár et al., "Behavior recognition via sparse spatio-temporal features," in *Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 65–72 (2005).
18. D. Lowe, "Object recognition from local scale-invariant features," in *Seventh IEEE Int. Conf. on Computer Vision*, Vol. **2**, 1150–1157, IEEE, Kerkyra (1999).
19. B. Herbert et al., "Speeded-up robust features (SURF)," *Comput. Vision Image Understanding* **110**, 346–359 (2008).
20. J. A. Hartigan and M. A. Wong, "Algorithm AS 136: a K-means clustering algorithm," *Applied Statistics* **28**(1), 100–108 (1979).
21. M. S. Ryoo and J. K. Aggarwal, "Spatio-temporal relationship match: video structure comparison for recognition of complex human activities," in *Int. Conf. on Computer Vision*, pp. 1593–1600, Kyoto, Japan (2009).
22. X. Sun, M. Chen, and A. Hauptmann, "Action recognition via local descriptors and holistic features," in *Computer Vision and Pattern Recognition*, pp. 58–65, Miami, Florida (2009).
23. A.-P. Ta et al., "Pairwise features for human action recognition," in *Int. Conf. on Pattern Recognition*, pp. 3224–3227, Istanbul, Turkey (2010).
24. C. Wu, A. H. Khalili, and H. Aghajan, "Multiview activity recognition in smart homes with spatio-temporal features," in *Int. Conf. on Distributed Smart Cameras*, pp. 142–149, Atlanta, Georgia (2010).
25. M. Bregonzio, T. Xiang, and S. Gong, "Fusing appearance and distribution information of interest points for action recognition," *Pattern Recognit.* **45**, 1220–1234 (2012).
26. Z. Zhang and D. Tao, "Slow feature analysis for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(3), 436–450 (2012).
27. L. Wiskott and T. Sejnowski, "Slow feature analysis: unsupervised learning of invariances," *Neural Comput.* **14**(4), 715–770 (2002).
28. L. Onofri and P. Soda, "Combining video subsequences for human action recognition," in *Int. Conf. on Pattern Recognition*, pp. 597–600, Tsukuba, Japan (2012).
29. M. Chen and A. Hauptmann, "MoSIFT: recognizing human actions in surveillance videos," Tech. Rep., Carnegie Mellon University Computer Science, Pittsburgh, Pennsylvania (2009).
30. N. Tishby, F. Pereira, and W. Bialek, "The information bottleneck method," in *Allerton Conf. on Communication, Control and Computing*, pp. 368–377 (1999).
31. M. Chen et al., "Action recognition using lie algebrized Gaussians over dense local spatio-temporal features," *Multimed. Tools Appl.* **74**(6), 2127–2142 (2015).
32. S. Singh, S. A. Velastin, and H. Ragheb, "MuHAVi: a multicamera human action video dataset for the evaluation of action recognition methods," in *Advanced Video and Signal Based Surveillance*, pp. 48–55 (2010).
33. K. Raja et al., "Joint pose estimation and action recognition in image graphs," in *IEEE Int. Conf. on Image Processing*, pp. 25–28, Brussels, Belgium (2011).
34. C. H. Hsieh, P. Huang, and M. D. Tang, "The recognition of human action using silhouette histogram," in *Australasian Computer Science Conf.*, Vol. 113, pp. 11–16, Perth, Australia (2011).
35. S. Cheema et al., "Action recognition by learning discriminative key poses," in *Int. Conf. on Computer Vision*, Barcelona, Spain, pp. 1302–1309 (2011).
36. S. Karthikeyan et al., "Probabilistic subspace-based learning of shape dynamics modes for multi-view action recognition," in *Int. Conf. on Computer Vision*, pp. 1282–1286, Barcelona, Spain (2011).
37. K. Guo, P. Ishwar, and J. Konrad, "Action recognition from video using feature covariance matrices," *IEEE Trans. Image Process.* **22**(6), 2479–2494 (2013).
38. A. Chaaraoui, P. Climent-Pérez, and F. Flórez-Revuelta, "Silhouette-based human action recognition using sequences of key poses," *Pattern Recognit. Lett.* **34**(15), 1799–1807 (2013).
39. S. Wang, K. Huang, and T. Tan, "A compact optical flowbased motion representation for real-time action recognition in surveillance scenes," in *IEEE Int. Conf. on Image Processing*, pp. 1121–1124, IEEE, Cairo, Egypt (2009).
40. I. N. Junejo and Z. A. Aghbari, "Using SAX representation for human action recognition," *J. Visual Commun. Image Represent.* **23**, 853–861 (2012).
41. S. Ji et al., "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(1), 221–231 (2013).
42. Z. Moghaddam and M. Piccardi, "Histogram-based training initialisation of hidden Markov models for human action recognition," in *Int. Conf. on Advanced Video and Signal Based Surveillance*, pp. 256–261, Boston, Massachusetts (2010).
43. Z. Moghaddam and M. Piccardi, "Training initialization of hidden Markov models in human action recognition," *Autom. Sci. Eng.* **36**(99), 1–15 (2013).
44. A. Tran et al., "Action recognition in the frequency domain," Cornell University Library, arXiv preprint: 1409.0908, Submitted on 2 September 2014.
45. A. Antonucci et al., "Robust classification of multivariate time series by imprecise hidden Markov models," *Int. J. Approximate Reasoning* **56**, 249–263 (2015).
46. P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-Based Surveillance Systems*, pp. 135–144, Springer-Verlag, New York (2002).
47. I. T. Jolliffe, *Principal Component Analysis*, 2nd ed., Springer-Verlag, New York (2002).
48. M. Blank et al., "Actions as space-time shapes," in *Int. Conf. on Computer Vision*, pp. 1395–1402, Beijing, China (2005).
49. C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *17th Int. Conf. on Pattern Recognition*, Vol. 3, pp. 32–36, Cambridge (2004).
50. D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Comput. Vision Image Understanding* **104**(2–3), 249–257 (2006).
51. R. Messing, C. Pal, and H. Kautz, "Activity Recognition using the velocity histories of tracked keypoints," in *Twelfth IEEE Int. Conf. on Computer Vision*, IEEE Computer Society, Washington, DC (2009).

52. A. Chaaraoui and F. Flórez-Revuelta, "Human action recognition optimization based on evolutionary feature subset selection," in *Genetic and Evolutionary Computation Conf.*, pp. 1229–1236, New York, New York (2013).
53. Y. Yan et al., "Multi-task linear discriminant analysis for multi-view action recognition," in *20th IEEE Int. Conf. on Image Processing*, pp. 2842–2846 (2013).
54. A. Farhadi and M. Tabrizi, "Learning to recognize activities from the wrong view point," *Lect. Notes Comput. Sci.* **5302**, 154–166 (2008).
55. R. Li and T. Zickler, "Discriminative virtual views for cross-view action recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2855–2862 (2012).
56. J. Liu et al., "Cross-view action recognition via view knowledge transfer," in *2011 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 3209–3216, IEEE, Providence, Rhode Island (2011).
57. B. Li, O. Camps, and M. Sznaier, "Cross-view activity recognition using hankelets," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1362–1369 (2012).
58. I. Junejo et al., "View-independent action recognition from temporal self-similarities," *IEEE Trans. Pattern Anal. Mach. Intell.* **33**, 172–185 (2011).
59. D. Weinland, E. Boyer, and R. Ronfard, "Action recognition from arbitrary views using 3D exemplars," in *IEEE 11th Int. Conf. on Computer Vision*, pp. 1–7 (2007).
60. A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Advances in Neural Information Processing Systems*, B. Schölkopf, J. C. Platt, and T. Hoffman, Eds., Vol. 19, pp. 41–48, MIT Press (2006).
61. C. H. Huang, Y. R. Yeh, and Y. C. Wang, "Recognizing actions across cameras by exploring the correlated subspace," *Lect. Notes Comput. Sci.* **7583**, 342–351 (2012).
62. K. Reddy, J. Liu, and M. Shah, "Incremental action recognition using feature-tree," in *IEEE 12th Int. Conf. on Computer Vision*, pp. 1010–1017 (2009).
63. X. Wu and Y. Jia, "View-invariant action recognition using latent kernelized structural SVM," *Lect. Notes Comput. Sci.* **7576**, 411–424 (2012).
64. A. Bobick and J. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.* **23**, 257–267 (2001).
65. I. Laptev et al., "Learning realistic human actions from movies," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1–8 (2008).

Marlon F. Alcantara received his PhD in computer science from the Institute of Computing at the University of Campinas, Brazil. He received his MSc in electrical engineering and his BSc in computer science from Santa Catarina State University, Brazil. His research interests include machine learning, computer vision, pattern recognition, and image processing.

Thierry P. Moreira is currently a PhD student in the Institute of Computing at the University of Campinas, Brazil. He received his MSc degree in computer science from the Institute of Computing at the University of Campinas, Brazil. He received his BSc in computer science from Amazonia University, Brazil. His research interests include machine learning, computer vision, pattern recognition, and image processing.

Helio Pedrini is currently a professor at the Institute of Computing at the University of Campinas, Brazil. He received his PhD in electrical and computer engineering from Rensselaer Polytechnic Institute, Troy, New York. He received his MSc in electrical engineering and his BSc in computer science, both degrees from the University of Campinas, Brazil. His research interests include image processing, computer vision, pattern recognition, machine learning, computer graphics, and scientific visualization.