Action Recognition From Videos using Sparse Trajectories

Alexandros Doumanoglou, Nicholas Vretos, Petros Daras

Centre for Research and Technology - Hellas (ITI-CERTH)
6th Km Charilaou - Thermi, Thessaloniki, Greece
{aldoum, vretos, daras}@iti.gr

Keywords: action recognition, videos, trajectories, classification

Abstract

In this paper, a novel, low-complexity, method for action recognition from videos is presented. A 3D Sobel filter is applied to the video volume resulting into a binary image with non-zero pixels in areas of motion. The non-zero valued pixels are spatially clustered using k-means and the most dominant centers of video motion are extracted. The centers are then tracked forming sparse trajectories, whose properties are later used to create a new feature type, namely the Histogram of Oriented Trajectories (HOT), describing the video. Feature vectors are finally passed to an AdaBoost classifier for classification. The proposed method reports competitive results in KTH and MuHAVi datasets, while remaining low in complexity and thus being suitable to be used in surveillance systems requiring low processing power.

1 Introduction

Nowadays, with the increasing demand on safety and security the need for intelligent surveillance systems is at its peak. Action recognition from videos consists one of the most important aspects of building intelligent surveillance systems and has attracted a lot of researchers of the computer vision community. Despite its popularity, the topic remains a challenging and complex one. Any effective and robust action recognition method has to face a number of difficulties such as non-stationary cameras, moving backgrounds as well as view-point and illumination variations. So far, action recognition has been addressed under various and distinct approaches resulting in varying outputs in both recognition accuracy and computational complexity. For surveillance systems in particular, where low-power and low-cost systems are often employed, algorithms of low computational complexity constitute a common requirement.

In this paper, we present a novel, low-complexity, method for action recognition in videos by extracting sparse trajectories of few representative centers of motion. A binary mask of pixels that correspond to areas of motion is firstly computed by applying a 3D Sobel filter to the input video. Subsequently, centers that represent the most dominant areas of motion are calculated by spatial clustering of the non zero valued pixels of the mask (i.e. the centers of the bigger clusters). These centers, called "centers of the most dominant motion", are then tracked forming sparse trajectories. Based on information from the sparse trajectories, we introduce a new feature type, namely the Histograms of Oriented Trajectories (HOT), which is subsequently passed to an AdaBoost [1] classifier for classification. The main novelty of this paper is the simplicity and low computational complexity of the method while retaining near state-of-the art results, making it suitable for surveillance systems requiring low-power consumption. We evaluate against the well known KTH dataset [12] achieving 88.7% recognition accuracy and the more challenging MuHAVi [13] dataset achieving recognition accuracy of 72.2% over 17 classes in total.

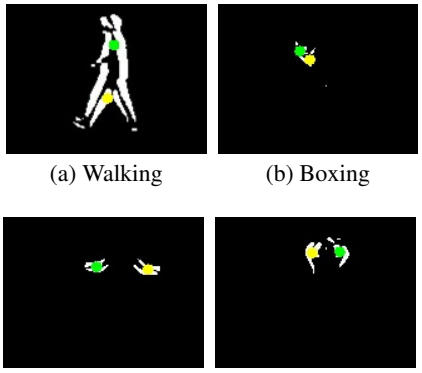
The rest of the paper is organized as follows: In Section 2 related work is discussed. In Section 3 the proposed approach is illustrated and detailed while in Section 4 evaluation results are given. Finally, Section 5 concludes the paper.

2 Related Work

Vision-based human action recognition, at its simplest form, can be regarded as a combination of feature extraction and subsequent classification of these features to predefined classes [9]. In [12], Schüldt *et al* represent motion patterns by using local space-time features [5] combined with SVM classification for human action recognition. In [4], Laptev *et al*, presented a method for video classification based on local space-time features, space-time pyramids and multi-channel non-linear SVMs. Later, Kovashka *et al* [3], proposed to learn the shapes of space-time feature neighborhoods that are most discriminative for a given action category.

In [14], dense points are sampled from each frame and then tracked based on displacement information from a dense optical flow field forming dense trajectories. HOGHOF and Motion Boundary Histogram (MBH) descriptors are then computed along these trajectories and used for action recognition. In [11], a high-level representation of a video is proposed based on many individual action detectors. Action recognition is carried out by SVM classifiers on the output of action detectors. This method reports highly accurate results on many datasets. However, the computational complexity of the method is several orders of magnitude higher than the rest of the methods reported in the bibliography, making it impractical for real life applications.

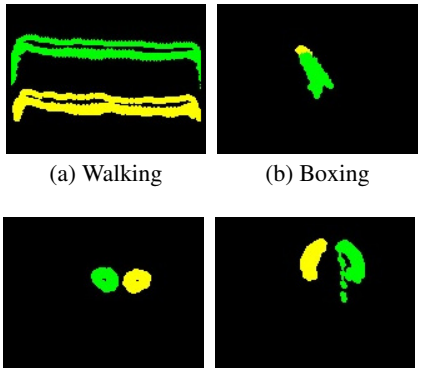
More recently, Oneata *et al* [8], focus on the low-level features and their encoding with application to action recognition



(a) Walking (b) Boxing

(c) Hand Clapping (d) Hand Waving

Figure 1. In green and yellow the 2 most dominant motion centers extracted by the k-means algorithm. Frames come from the KTH dataset.



(a) Walking (b) Boxing

(c) Hand Clapping (d) Hand Waving

Figure 2. In green and yellow the 2 most dominant motion center's trajectories. The sequences come from the KTH dataset.

from uncontrolled videos. The authors use the Fisher Vectors as an alternative to bag-of-words histograms in order to aggregate a small set of state-of-the-art low-level descriptors, in combination with linear classifiers. Very good results are also reported while using fewer features and less complex models. In [15], Wang *et al*, improve the method of [14] by taking into account camera motion and applying motion corrections/compensation. Camera motion estimation is performed by employing SURF descriptors and dense optical flow while the motion correction is applied to trajectories and improves the motion-based descriptors HOF and MBH. Finally, Liu *et al* [6] propose a method for action recognition by feature selection using the AdaBoost algorithm and classification by the naive Bayes nearest-neighbor classifier. Feature extraction is based on 3D-SIFT and 3D-HOG descriptors computed on overlapping sub-blocks of the video sequence.

3 Proposed Method

From a high level perspective, the proposed algorithm follows the standard feature extraction and classification workflow. Initially, the most dominant centers of motion are detected and subsequently tracked and registered in a frame-by-frame basis in order to form trajectories. Then, the trajectories' velocities and accelerations are computed and finally, the histogram of oriented trajectories (HOT) is built, forming the feature vector that is later used for classification.

The input to the algorithm is considered to be a video sequence. Firstly, the video sequence is transformed to a 3D volume considering the 2 spatial dimensions of the frame as the first two dimensions of the 3D volume and the third dimension expanding in time. Thus, a volume is created containing along the third dimension all the video frames. Then a 3D Sobel filter is applied to this volume. The 3D Sobel filter will result in edges along the third dimension, which can be interpreted as trajectories of moving pixels. Thus, the 3D Sobel filter serves as a means to extract the motion information from the video. For each time frame t, the Sobel filter's output is thresholded and result to a binary image where the non-zero valued pixels indicate areas of motion. Thereafter, we choose to have k centers of dominant motion based on the videos' dataset. These k centers are being calculated by applying the k-means clustering algorithm [7] in the spatial image domain and only for the pixel coordinates of the non-zero valued pixels of the binary image (see Fig. 1). Subsequently, these k motion centers are being tracked, forming k distinct and sparse motion trajectories that are later used to form features describing an action.

Since k-means is performed at each time frame t, motion centers need to be identified and registered in a frame-by-frame manner. To do so, we employ a nearest neighbor strategy to match each motion center of frame t to the frame $t-1$ and thus, track centers and form trajectories. In practice, matching the motion centers of frame t to motion centers of frame $t-1$ is prone to errors when the motion centers are spatially close to each other. To resolve the latter, we use a more robust technique employing the moving average throughout the trajectory. Let $\mathbf{c}^i(t), i \in \{0, ..., k-1\}$ denote the i-th motion center extracted by the k-means algorithm for frame t. It should be clear that up to this point, $\mathbf{c}^i(t)$ has not been registered to a particular trajectory and is still unidentified. To perform identification and registration let $\mathbf{c}_j(t)$ denote the motion center registered with trajectory $T_j, j \in \{0, ..., k-1\}$ at frame t. Superscript i in $\mathbf{c}^i(t)$ is used to denote a yet unidentified motion center while the subscript j in $\mathbf{c}_j(t)$ denotes a motion center registered to trajectory T_j. To increase trajectory extraction robustness, for each trajectory T_j and at time t we monitor its motion center's moving average position, $\bar{\mathbf{c}}_j(t)$ using a window of w frames, i.e. $\bar{\mathbf{c}}_j(t) = \frac{1}{w}\sum_{n=t-w}^{n=t-1}\mathbf{c}_j(n)$. Then, at frame t the unidentified motion center $\mathbf{c}^i(t)$ resulted from the output of the k-means algorithm, is assigned to the trajectory T_j only if $||\mathbf{c}^i(t) - \bar{\mathbf{c}}_j(t)||_2$ is minimal, with $||\cdot||_2$ denoting the euclidean norm. Thus, $\mathbf{c}_j(t) = \arg\min_{\mathbf{c}^i(t)}(||\mathbf{c}^i(t) - \bar{\mathbf{c}}_j(t)||_2)$. Figure 2 depicts trajectories extracted using this method.

By the completion of the previous step, the output consists of k trajectories, which characterize the most dominant motions inside the video. Each such trajectory T_j can be expressed as a $N \times 2$ matrix $\mathbf{M}_j$, with each row containing the (x, y) spatial coordinate pairs of the j-th motion center for each frame $t \in \{0, ..., N-1\}$, where N denotes the number of all

frames in the video. Furthermore, let W and H denote the frame's width and height, respectively. To achieve invariance across different video resolutions, we normalize the coordinate pairs to $[0, 1]$ by dividing each (x, y) coordinate by the factor $s = \max(W, H)$ to preserve the original aspect ratio. Finally, we smooth the trajectories by applying a moving average filter to the matrix $\mathbf{M}_j$.

Based on the extracted trajectories T_j, $j \in \{0, ..., k-1\}$, we can compute the velocities for each trajectory at time t as $\mathbf{v}_j(t) = \mathbf{M}_j(t) - \mathbf{M}_j(t-1)$, where $\mathbf{M}_j(t)$ is the t-th row of matrix $\mathbf{M}_j$. Moreover, the trajectory accelerations at time t can be computed as: $\mathbf{a}_j(t) = \mathbf{v}_j(t) - \mathbf{v}_j(t-1)$. After we have calculated $\mathbf{v}_j(t)$ and $\mathbf{a}_j(t)$, we use them in order to extract the feature vector that is used for video classification. The proposed feature vector, namely Histogram of Oriented Trajectories (HOT), is based on the concatenation of 3 different histograms: 1) the joint histogram, with respect to the different centers, of velocity orientations, 2) the joint histogram, with respect to the different centers, of velocity magnitudes and 3) the joint histogram, with respect to the different centers, of acceleration magnitudes.

To calculate the joint histogram of velocity orientations, we define the k dominant centers' marginal histograms having all NB_1 distinct bins. Then, the joint histogram will have $(NB_1)^k$ bins resulting from the NB_1 permutations of k with repetitions. To quantize the orientations we correspond each one of the NB_1 bins to a different orientation interval, equally distributed in the range $0 - 360^o$. For all j, each velocity vector $\mathbf{v}_j(t)$ is assigned a quantization index $idx\_or_j(t) \in \{0, ...NB_1 - 1\}$ of the corresponding angle. Finally, all the quantization indices $idx\_or_j(t)$ contribute to an unweighted vote to the joint histogram's bin indexed by $(NB_1)^{k-1} idx\_or_{k-1}(t) + (NB_1)^{k-2} idx\_or_{k-2}(t) + ... + idx\_or_0(t)$, assuming 0-based indexing. In other words, all velocity vectors at frame t from the different motion centers contribute together to a single bin of the joint histogram of velocity orientations, thus encoding the interplay between the dominant motions trajectories.

The computation of the joint histograms of velocity magnitudes and acceleration magnitudes follows the same methodology, with the difference that instead of quantizing the vector orientations, quantization of the magnitude of the vectors is assumed. Following the previous discussion, let NB_2 and NB_3 denote the number of distinct bins for velocities' and accelerations' magnitude, respectively. Moreover, let each velocity magnitude and acceleration magnitude be a clamped value in the interval $[0, \max_{\mathrm{v}}]$ and $[0, \max_{\mathrm{a}}]$, correspondingly. Furthermore, for the shake of clarity and in the current context, let $\mathbf{v}_j^m(t)$ denote the velocity vector of the j-th motion center at frame t and for the video $m \in \{0, ..., L-1\}$, where L denotes the total number of videos in the training set. Then,

$$\max_{\mathrm{v}} = \frac{1}{kL} \sum_{j,m} \max_t (||\mathbf{v}_j^m(t)||_2) \tag{1}$$

Similarly,

$$\max_{\mathrm{a}} = \frac{1}{kL} \sum_{j,m} \max_t (||\mathbf{a}_j^m(t)||_2) \tag{2}$$

Thus, for velocities, the magnitude of each vector $||\mathbf{v}_j(t)||_2$ is mapped to an index by the following formula (assuming 0-based indexing):

$$idx\_v_j(t) = \min(\lfloor \frac{||\mathbf{v}_j(t)||_2}{\max_{\mathrm{v}}} NB_2 \rfloor, NB_2 - 1) \tag{3}$$

Likewise, for accelerations, the magnitude of each vector $||\mathbf{a}_j(t)||_2$ is mapped to the following 0-based index:

$$idx\_a_j(t) = \min(\lfloor \frac{||\mathbf{a}_j(t)||_2}{\max_{\mathrm{a}}} NB_3 \rfloor, NB_3 - 1) \tag{4}$$

Obviously, $idx\_v_j(t) \in \{0, ..., NB_2 - 1\}$ and $idx\_a_j(t) \in \{0, ..., NB_3 - 1\}$. Finally, the velocity magnitudes of the k most dominant motion centers altogether contribute to the histogram bin indexed by:

$$(NB_2)^{k-1} idx\_v_{k-1}(t) + (NB_2)^{k-2} idx\_v_{k-2}(t) + ... + idx\_v_0(t) \tag{5}$$

while for acceleration magnitudes the previous equation holds in a similar manner, where NB_2 is substituted with NB_3 and $idx\_v_j(t)$ with $idx\_a_j(t)$.

4 Experimental Results

In this section, the evaluation of the proposed algorithm is presented. Firstly, the evaluation is carried out on the KTH dataset [12] that contains 600 videos in total from 25 people performing 6 different actions. Secondly, we evaluate against the MuHAVi dataset [13] which contains 17 actions all performed by 7 people and captured by 8 cameras.

In order to choose the parameters of the system, we run a grid based approach resulted in the proposed values. Firstly, the number of most dominant motion centers per video was set to $k = 2$. This means that for each video, 2 trajectories of the most dominant motion centers are extracted. Secondly, for trajectory extraction the parameter w is set to 25. Finally, the HOT parameters NB_1, NB_2 and NB_3 are all set to 8 and the classifier used was AdaBoost.

For evaluation in KTH dataset we use the standard procedure proposed in [12]. The total recognition accuracy achieved by our method is 88.7%. The resulted confusion matrix is illustrated in table 2, while a comparison with respect to some methods in the bibliography is given in table 3. Most of the methods in the bibliography, like [10] and [4], lack a computational complexity analysis except for [11] and are not included in the table, but even [11] is proved to be a too complex algorithm to run in systems requiring low processing power.

While MuHAVi is a multi-view dataset, we use a single view for each action and more precisely the view which is perpendicular to the action. The evaluation strategy is set to leave-one-actor-out. The total recognition accuracy of the proposed

	Climb Ladder	Jump Over Fence	Punch	Crawl On Knees	Jump Over Gap	Run Stop	Drunk Walk	Kick	Shotgun Collapse	Pickup Throw Object	Pull Heavy Object	Smash Object	Walk Turn Back	Wave Arms	Draw Graffiti	Look In Car	Walk Fall
Climb Ladder	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Jump Over Fence	0	**85.7**	0	0	0	0	14.3	0	0	0	0	0	0	0	0	0	0
Punch	0	0	**42.8**	0	0	0	0	14.3	14.3	28.6	0	0	0	0	0	0	0
Crawl On Knees	0	0	0	**100**	0	0	0	0	0	0	0	0	0	0	0	0	0
Jump Over Gap	0	14.3	0	0	**57.1**	0	14.3	14.3	0	0	0	0	0	0	0	0	0
Run Stop	0	0	0	0	0	**71.4**	0	0	0	0	0	0	28.6	0	0	0	0
Drunk Walk	0	0	0	0	0	0	**100**	0	0	0	0	0	0	0	0	0	0
Kick	0	14.3	0	0	0	0	14.3	**57.1**	0	14.3	0	0	0	0	0	0	0
Shotgun Collapse	0	0	0	0	0	0	0	0	**100**	0	0	0	0	0	0	0	0
Pickup Throw Object	0	0	28.6	0	0	0	14.3	0	0	**14.3**	0	42.8	0	0	0	0	0
Pull Heavy Object	0	0	0	0	0	0	14.3	0	0	0	**85.7**	0	0	0	0	0	0
Smash Object	0	0	0	0	0	0	0	0	14.3	14.3	0	**71.4**	0	0	0	0	0
Walk Turn Back	0	0	0	0	0	0	0	0	0	0	0	0	**85.7**	0	0	0	14.3
Wave Arms	0	0	0	0	0	0	0	0	0	0	0	0	0	**71.4**	28.6	0	0
Draw Graffiti	0	0	0	0	0	0	0	0	0	14.3	0	0	0	14.3	**42.8**	28.6	0
Look In Car	28.5	0	0	0	0	0	0	0	0	0	0	0	0	0	28.5	**42.8**	0
Walk Fall	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**100**

Table 1. 17-class MuHAVi dataset confusion matrix.

	Walk	Jog	Run	Box	Hclp	Hwav
Walk	**92**	2.9	0.8	4.3	0	0
Jog	7.6	**79.0**	12.7	0.2	0.5	0
Run	1.5	15.9	**81.9**	0.3	0.2	0.2
Box	0.1	0	0	**96.6**	3	0.3
Hclp	0	0	0	8.4	**88.6**	3
Hwav	0.1	0	0	3.1	2.2	**94.6**

Table 2. KTH dataset confusion matrix.

Method	Accuracy (%)
Schüldt et al [12]	71.7
Klaser et al [2]	84.3
Proposed Method	**88.7**
Sadanand et al [11]	98.2

Table 3. Accuracy of the methods in the bibliography on KTH dataset

method for all the 17 classes is 72.2% and the confusion matrix is given in table 1. We further evaluate on a subset of 10 action classes of the same dataset, achieving 92.8% recognition accuracy while the confusion matrix is given in table 4.

As for the computational complexity aspects of the proposed method, we run our MATLAB code on an Intel i7-2700K 3.5GHz CPU with 8GB RAM. In table 5, the average per video processing times for videos from the KTH dataset are given. The average video length in KTH dataset is 484 frames and the average per video processing time of our algorithm is measured to be 1.4 sec.

	Climb Ladder	Crawl On Knees	Run Stop	Drunk Walk	Shotgun Collapse	Pull Heavy Object	Smash Object	Walk Turn Back	Wave Arms	Walk Fall
Climb Ladder	**100**	0	0	0	0	0	0	0	0	0
Crawl On Knees	0	**100**	0	0	0	0	0	0	0	0
Run Stop	0	0	**71.5**	0	0	0	0	28.5	0	0
Drunk Walk	0	0	0	**85.7**	0	0	0	14.3	0	0
Shotgun Collapse	0	0	0	0	**100**	0	0	0	0	0
Pull Heavy Object	0	14.3	0	14.3	0	**71.4**	0	0	0	0
Smash Object	0	0	0	0	0	0	**100**	0	0	0
Walk Turn Back	0	0	0	0	0	0	0	**100**	0	0
Wave Arms	0	0	0	0	0	0	0	0	**100**	0
Walk Fall	0	0	0	0	0	0	0	0	0	**100**

Table 4. 10-class MuHAVi dataset confusion matrix.

Stage	**Time (sec)**
Trajectory extraction	1.4
HOT feature extraction	0.006
AdaBoost Testing	0.01
Total	1.416

Table 5. Average per video processing times, for videos from the KTH dataset.

5 Conclusion

In this paper, a novel method for action recognition from videos has been presented based on sparse trajectories. Moreover, a new feature vector type has been proposed for classification: the histogram of oriented trajectories (HOT), which is based on properties of the aforementioned trajectories, particularly on trajectory orientations and velocity/acceleration magnitudes. Evaluation of the algorithm was conducted in KTH [12] and MuHAVi [13] datasets showing competitive results. The proposed algorithm is low in computational complexity and thus suitable to be used in surveillance systems requiring low processing power.

Acknowledgements

The research leading to these results has been supported by the EU funded project FORENSOR (GA 653355).

References

[1] Yoav Freund, Robert Schapire, and N Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[2] Alexander Kläser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conference*, pages 995–1004, sep 2008.

[3] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2046–2053, June 2010.

[4] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[5] Ivan Laptev and Tony Lindeberg. Space-time interest points. In *in ICCV*, pages 432–439, 2003.

[6] Li Liu, Ling Shao, and Peter Rockett. Human action recognition based on boosted feature selection and naive bayes nearest-neighbor classification. *Signal Process.*, 93(6):1521–1530, June 2013.

[7] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif., 1967. University of California Press.

[8] D. Oneata, J. Verbeek, and C. Schmid. Action and event recognition with fisher vectors on a compact feature set. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1817–1824, Dec 2013.

[9] Ronald Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976 – 990, 2010.

[10] M.S. Ryoo and J.K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1593–1600, Sept 2009.

[11] Sreemanananth Sadanand and Jason J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, pages 1234–1241. IEEE Computer Society, 2012.

[12] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local svm approach. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03*, ICPR '04, pages 32–36, Washington, DC, USA, 2004. IEEE Computer Society.

[13] 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 (AVSS), 2010 Seventh IEEE International Conference on*, pages 48–55, Aug 2010.

[14] Heng Wang, A. Klaser, C. Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176, June 2011.

[15] Heng Wang and C. Schmid. Action recognition with improved trajectories. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3551–3558, Dec 2013.