Real-Time Human Action Recognition Using Full and Ultra High Definition Video

Gloria Castro-Muñoz and Jorge Martínez-Carballido
Department of Electronics Science, Instituto Nacional de Astrofísica Óptica y Electrónica
Puebla, México, 72840
Email: cgloria@inaoep.mx; jmc@inaoep.mx

***Abstract*—Given that camera products are moving rapidly towards Ultra High Definition in video surveillance area, this work presents a real-time human action recognition method based on simple and clear concepts which achieves a great *Accuracy-Speed* performance on videos up 8K Ultra High Definition. The method was evaluated on *i3DPost* dataset at different video resolutions. The comparative analysis shows that our method outperforms all state of the art methods, in accuracy (99%) and to the best of our knowledge; it is the first one that shows real-time performance on videos up 8K UHD.**

Keywords- Human Action Recognition; Real-Time Performance; Ultra High Definition Video.

I. INTRODUCTION

The increasing demands for security and safety by society have led the rapid evolution of video surveillance technology based on computer vision. Development started with the introduction of pixel surveillance camera, then megapixels camera, after HD (High Definition) and Full HD cameras and today UHD (Ultra High Definition) surveillance camera.

The growing need for higher image resolution in video surveillance is a consequence of the need to improve the quality of footage. Higher video definition provides clearer images and crisper videos to identify and monitor actions of criminals readily and generate images that can be used as irrefutable evidence against criminals in a court law.

With the ever increasing of image quality, video surveillance benefits by easing criminal actions detection; but on the other hand, higher quality image generates higher computational and communication load and hence the need to propose new and better solutions for High definition video analytics. This is the main purpose of our work.

The research topic of Human Action Recognition (HAR) from streaming video has gathered a great number of publications [1]. However, very few methods reported in those publications are applicable for real-time processing and a very smaller percentage of the HAR methods performed in real-time used HD videos. The great majority of the research is focused on obtaining high recognition rates without giving relevance to computational effort, speed and constant increase in image quality.

Among methods reporting real-time operation with any timing evaluation on publicly available datasets are: methods in [2,3,4,5] use low resolution videos (180x144) with a maximum processing rate of 98 fps; methods in [6,7,8] used videos of low and medium resolution (180x144 and 720x576) with a speed performance of 263 fps and 94 fps respectively; Experimental results in HD (1024x764) reported in [9] achieve a processing rate of 50fps; in [10] the method was evaluated in low, medium and HD resolution videos with a maximum processing rate of 32 fps.

Of the above methods only two [9,10] have been tested with HD videos showing real-time performance. Drawn from this analysis and motivated to give new solutions with real-time performance for full HD videos, we present a real-time HAR method that achieves a high *Accuracy-Speed* performance on videos up to 8K UHD.

II. PROPOSED METHOD

The main steps of the method are illustrated in Fig. 1. The algorithm begins with a video sub-sequence (snippet) of *n* frames containing binary silhouettes. Then, for each snippet frame, a *tracker* locates the smallest rectangle that encapsulates the human silhouette (*Bounding Box* BB). Then, the human body is represented using only two rectangular boxes, the original BB, and one smaller rectangle contained within this (*Knee Box* KB). After, two types of features are extracted, *local features* and *global features*. We have adopted this nomenclature in order to highlight that *global features* characterize a full snippet; while *local features* characterize each frame. The local features are four and they describe the morphology of a silhouette; and global features are five and they describe how movement unfolds. After that, the global features are concatenated into a single *action feature vector* per snippet. Finally, the action feature vector is fed to a *hierarchical system of linear classifiers* which discriminates among 6 actions and assigns an *action label* to the corresponding snippet entry.

A. Tracking and pose representation

The BB tracking is an algorithm proposed by ours to track the human silhouette frame by frame in a simple and fast way. The main objective of the algorithm is reducing the search area by assumption that the human silhouette on each frame should be located in a neighboring region with regard to the region where it was located on a previous frame. The algorithm consists of the next steps. At the outset

978-1-4673-9795-7/15 $31.00 © 2015 IEEE
DOI 10.1109/CSCI.2015.12

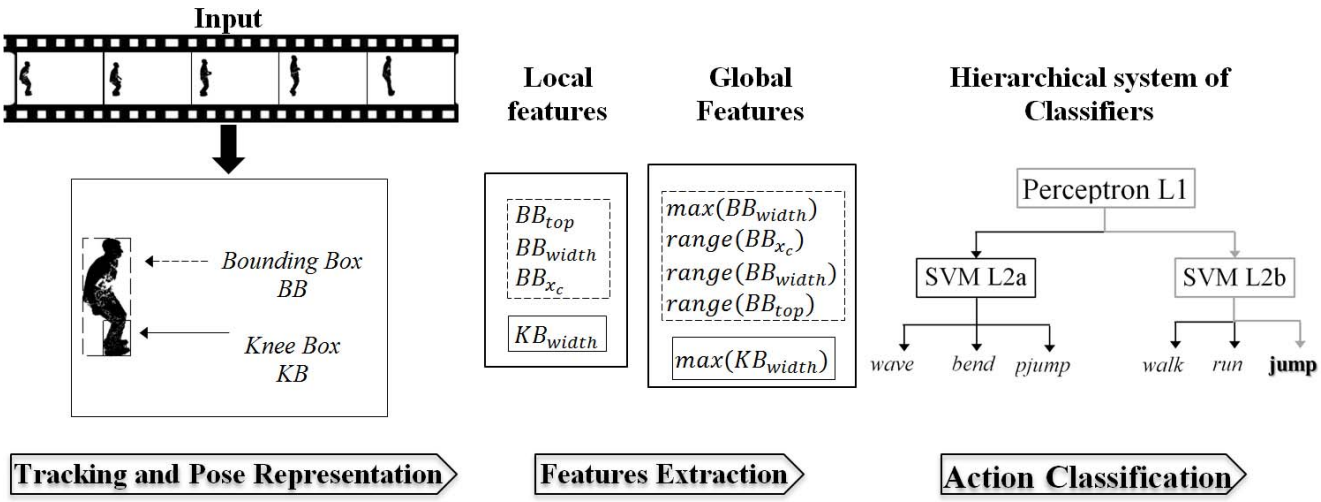

Figure 1. Overview of the proposed method.

(frame 1), the tracker performs a search for the silhouette inside the *initial BB search area* χ_1 which is the whole initial frame and thus obtaining the *first Bounding Box* BB_1. Next, in subsequent frames ($n \geq f > 1$), an estimated *BB search area* $\widetilde{\chi_f}$ is extended in ω pixels on the four edges of the previous BB (BB_{f-1}) to reduce the tracker space of search. Where ω is a configurable parameter and it represents the maximum displacement of the silhouette between two consecutive frames, it is computed with (1) for the fastest analyzed activity. After, once the estimated *BB search area* $\widetilde{\chi_f}$ was obtained, the tracker scans within this area to get the current BB (BB_f). Finally, steps two and three are repeated for the remaining frames.

$$\omega = \left\lceil \left(\frac{1}{fps}\right) * \begin{pmatrix} avg\ action \\ speed \end{pmatrix} * \left(\frac{1\ pixel}{meters\ per\ pixel}\right) \right\rceil \quad (1)$$

One of the most important elements in the HAR system presented in this work is "*the human body representation*". This is because the proposed human model allowed us to select a feature set to represent actions which are clear, reduced, and easy to compute. Thus, with this representation, the main objective is reduce the complex human form into a less detailed one that still retain enough information to distinguish the selected actions set. Therefore, based on the analysis of the human body movement involved in each of the actions to recognize, it was concluded that only two rectangular boxes were enough to model the human figure; *Bounding Box* (BB), and *Knee Box* (KB). An example of this representation for action "jump" is shown in Fig. 1, on stage *"Tracking and Pose Representation"*. The first box (dashed line) covers 100% of the silhouette (full body) and the second box (solid line) covers 30% of the lower part of the BB (knees, legs and feet). The percentage for Knee Box was selected according to standard geometrical proportions of human body [11].

B. Features Extraction

Two types of features are computed in this work: a) *Local features* that describe the morphology of the human body and b) *Global Features* that describe how movement unfolds.

The *local features* are four and they are computed for each frame, three for BB (width, centroid abscissa coordinate and upper edge coordinate) and one feature on KB (width). The features are computed using rectangular coordinates in equations (2) to (5). The four edge coordinates for *BB* are: left (x_{lft}), right (x_{rgt}), top (y_{top}), bottom (y_{btm}) and coordinates $k_{lft}, k_{rgt}, k_{top}, y_{btm}$ define *KB*.

Bounding Box

Width:

$$BB_{width} = |x_{rgt} - x_{lft}| \quad (2)$$

Centroid abscissa:

$$BB_{x_c} = x_{lft} + \left|\frac{x_{rgt} - x_{lft}}{2}\right| \quad (3)$$

Upper edge coordinate:

$$BB_{top} = y_{top} \quad (4)$$

Knee Box

Width:

$$KB_{width} = |\, k_{rgt} - k_{lft}| \quad (5)$$

The *global features* are five and they are representative of one snippet. Four of them are extracted from BB (maximum width, range width, maximum horizontal displacement and maximum vertical scrolling), and one from KB (maximum width). The five global features (G_F) are computed per snippet through an analysis performed on all the n frames. Considering that i is the frame index and that a snippet has n frames, global features are computed according to:

Bounding Box

Maximum width:

$$G_{F1} = \max_{i \in \{1,..,n\}} (BB_{width}[i]) \quad (6)$$

Maximum horizontal shift:

$$G_{F2} = \operatorname{range}_{i \in \{1,..,n\}} \left(BB_{x_c}[i]\right) \quad (7)$$

Range width:

$$G_{F3} = \operatorname{range}_{i \in \{1,..,n\}} (BB_{width}[i]) \quad (8)$$

Maximum vertical shift:

$$G_{F4} = \operatorname{range}_{i \in \{1,..,n\}} \left(BB_{top}[i]\right) \quad (9)$$

Knee Box (Lowest 30%)

Maximum width :

$$G_{F5} = \max_{i \in \{1,..,n\}} (KB_{width}[i]) \quad (10)$$

Finally the computed global features are concatenated to get a single action feature vector per snippet:
$\boldsymbol{v} = [G_{F1},\ G_{F2},\ G_{F3},\ G_{F4},\ G_{F5}]^T$.

C. Action Classification

The classification task in this work is carried out by the hierarchical system of classifiers shown in *"Action Classification"* stage on Fig. 1. This classification model captures the hierarchical nature of the set of human actions proposed and organizes this set in classes inside classes, allowing us to split the largest classification task in decision making processes less complicated.

The hierarchical system of classifiers consists of two levels of classification. At the first level of classification the perceptron L1 (Level 1) is responsible for determining the class of the input action feature vector between two classes: those actions with displacement $\mathcal{C}_1 = \{walk, run, jump\}$ and those actions happening at the same position $\mathcal{C}_2 = \{wave, bend, pjump\}$. Depending on the class assigned at the first level, one of the two SVMs at the second level of classification is enabled, *SVM L2a* (Level L2) and *SVM L2b* (Level 2). *SVM L2a* assigns a class among three possible classes $\mathcal{C}_{1,a} = \{wave\}$ $\mathcal{C}_{1,b} = \{bend\}$ and $\mathcal{C}_{1,c} = \{pjump\}$. On the other hand, *SVM L2b* assigns one of three possible classes $\mathcal{C}_{2,a} = \{walk\}$ $\mathcal{C}_{2,b} = \{run\}$ and $\mathcal{C}_{2,c} = \{jump\}$.

Each classifier is independently training using a supervised learning model. The learning model use a set of m training samples obtained from each dataset, where each sample is a pair $\{(\boldsymbol{x}_i, d_i)\}_{i=1}^{m}$ consisting of an feature vector ($\boldsymbol{x}_i$), with 2 to 4 entries, which is a sub-set of the set of five global features and a desired output value (action label d_i). Only two *global features* comprising the feature vector feed to ***perceptron L1***, G_{F2} and G_{F3}; ***SVM L2a*** uses the feature vector $[G_{F1}, G_{F3}, G_{F4}, G_{F5}]^T$, and ***SVM L2b*** uses feature vector $[G_{F3}, G_{F4}, G_{F5}]^T$.

The perceptron used in this work is a binary classifier based in the Rosenblatt model [12], which consisting of a single neuron and the two SVMs are multi-class classifiers with linear kernels based in the multi-class Bias SVM (BSVM) formulation described in [13]. We had choose a perceptron at the first level of classification because the two classes to separate at this level are linearly separable with a very margin of separation and then, a perceptron is computational attractive because of its simplicity. The reasons of using SVMs at second level of classification are: 1) The SVM has good generalization ability and 2) Linear hyper-planes have small room between classes at this level so we need classifiers that maximizes the margin of separation between classes, as it happen in a SVM.

III. Experimental Results

In order to verify the effectiveness of our method, experiments were conducted on i3DPost public dataset [14]. Section *A* shows the accuracy evaluation. Then, in section *B*, we assess the algorithm speed performance and finally, in section *C*, we compare our results with other state-of-the-art methods.

A. Accuracy Evaluation

The method was tested on i3DPost dataset using Leave-One-Out Cross-Validation (LOOCV) protocol [15]. The i3DPost dataset is a Full-HD (High Definition) resolution video dataset (1920 x 1080, 25 fps) showing eight different persons with each person performing 12 different human motions (six actions and six interactions). The subjects have different sex, nationality, and significant differences in body sizes, and clothing. Some examples of frames from the i3DPost dataset are shown in Fig. 2. Our experiments were conducted on the subset of six actions {"run," "walk," "jump", "wave1", bend and "pjump"} using two points of view.

Silhouettes were obtained by applying background subtraction and the Otsu thresholding method [16] on the blue channel.

In order to visualize and measure the performance of HAR algorithm on i3DPost dataset; we obtained the *confusion matrix* shown in Fig. 3. It is observed that the *correct classification rate* (CCR) using the LOOCV protocol is 99%. According to this confusion matrix, walk and run are the only two actions confused by the proposed method. This is due to the fact that some actors do fast walk instead of running in the dataset.

Figure 2. Examples of frames extracted from video sequences of the i3DPost dataset.

	walk	run	jump	bend	wave	pjump
walk	**100.0**	0.0	0.0	0.0	0.0	0.0
run	6.3	**93.8**	0.0	0.0	0.0	0.0
jump	0.0	0.0	**100.0**	0.0	0.0	0.0
bend	0.0	0.0	0.0	**100.0**	0.0	0.0
wave	0.0	0.0	0.0	0.0	**100.0**	0.0
pjump	0.0	0.0	0.0	0.0	0.0	**100.0**

Figure 3. Confusion Matrix obtained for i3DPost dataset using LOOCV protocol.

B. Timing evaluation

In this section, we compute the algorithm time performance at different video resolutions. First, we rescaled the i3DPost video size from 1/8 to 4 times obtaining video resolutions from 32kpix to 33Mpix. Then, the average run time per snippet was computed by adding partial execution time for each stage of the proposed method.

The algorithm was implemented in Microsoft Visual C++ 2010 Express of 32 bits using the OpenCV library [17]. Performance was measured on a notebook with Windows 7, an Intel iCore7-3610QM microprocessor at 2.3 GHz, 6 GB RAM, a SATA-3 hard drive, and DTR 300 Mbps.

The average run time per frame was computed with (11). It was obtained by adding partial execution time for each stage of the proposed method.

$$Avg\ run\ time_{per\ frame} = tracking_{\ time} + LF_{time} + GF_{time} \quad (11)$$

where $tracking_{time}$ is the average run time to extract *Bounding Boxes*, LF_{time} is the average run time for *local feature extraction*, and GF_{time} is the average run time for *global feature extraction*.

Temporal evaluation results for different i3DPost dataset resolutions are shown in Table I. Results show that despite the algorithm was evaluated in Full HD, 4K UHD and 8K UHD video resolution; our algorithm still shows real-time performance with a processing rate from *126,613 fps* in low resolution videos to *46fps* in 8K UHD video resolution.

TABLE I. TIME PERFORMANCE FOR DIFFERENT I3DPOST DATASET RESOLUTIONS

	240x135	480x270	960x540	1920x1080	3840x2160	7680x4320
run time per frame (ms)	0.008	0.013	0.081	0.319	4.76	21.32
Processing rate (fps)	126,613	76,661	12,399	3,138	210	46

Bar graph in Fig. 4 shows the contribution percentages for each stage to average run time per frame. It is observed that as frame size increases, percentage contribution of local features extraction run time decreases and the contribution of tracking grows. Finally, we must emphasize that in HD video resolution, contribution of each stage becomes steady.

Stage largest contributor to total average run time per frame is tracking, which contributes over 98% on all different resolutions.

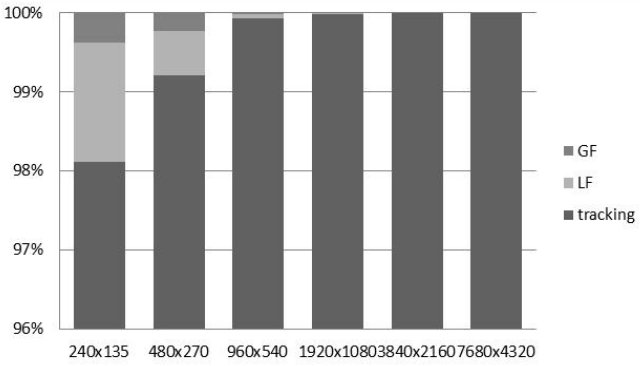

Figure 4. Contribution percentages of each stage to average run time for different i3DPost dataset resolutions.

C. Comparison with other methods

This section presents a comparison of the proposed method with other state-of-the-art methods in terms of *accuracy* (CCR) and *processing speed* (fps).

TABLE II. COMPARISON OF OUR METHOD WITH OTHER METHODS ON THE I3DPOST DATASET.

Approach	Year	Input	Actions	Features	Protocol	CCR (%)	FPS
Gkalelis *et al.* [21]	2009	Silhouettes downsized to W x H	5	W x H[a]	LOSO	90	N/A
Holte *et al.* [20]	2011	Raw Images	6	2304xU[b]	LOAO	89.58	N/A
Iosifidis *et al.* [18]	2012	Silhouettes dounsized to 64 x 64	6	4096	LOAO	95.33	N/A
Iosifidis *et al.* [19]	2013	Silhouettes downsized to 32 x 32	6	1024	LOAO	98.16	N/A
Our method	**2015**	**Raw Silhouettes**	**6**	**4**	**LOAO**	**99.0**	**3,138**

a. Where W is the width and H is the height of the smallest BB.
b. Where U is the number of bins in radial direction.

Methods were evaluated using the same dataset (i3DPost dataset) and protocol (LOOCV). According to Table II, our method is the only one which reports quantitative results in terms of processing rate, managing to process a maximum of *3,138* frames per second. In terms of accuracy, our method gets higher accuracy than the other methods. In terms of computational effort, the proposed method uses only arithmetic operations to compute the features for frame (addition, subtraction, multiplication, division); it requires up *1024* times less features than the other methods and it does not need to downsize the silhouette to reduce the computational effort as it happens in other methods (0.4% of average original ROI area or 250 times area downsize) and it does not require additional techniques to improve silhouettes quality. Then, our method achieves a better *overall performance* in compassion with other methods, where *"overall performance"* means to achieve a great *Accuracy–Speed performance*, which in our case it significantly improves both speed and accuracy to previous works.

IV. CONCLUSION

In this paper, we presented a method for human action recognition which combines very simple technics together with clear and simple concepts to achieve a high performance in terms of recognition and execution time. The method uses bounding box estimation, a simple pose representation based on 2 rectangular boxes, a reduced number of features (four per frame) which are easy to compute (based on simple arithmetic operations), and linear classifiers. Experimental results on the i3DPost dataset show the presented method achieves higher accuracy (99%) with far fewer features (up to 1024 times less features) than state of the art methods, and high processing rates on low, high, Full HD and 8K Ultra HD video resolution (126,613fps, 12,399fps 3,138 fps, 46fps). Then, our method achieves a high *Accuracy-Speed* performance which is essential in a large part of HAR application areas.

REFERENCES

[1] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, p. 16, 2011.

[2] J. Hernández, A. Montemayor, J. Pantrigo, and A. Sánchez, "Human action recognition based on tracking features," in *Foundations on Natural and Artificial Computation*, La Palma, Canary Islands, Spain, 2011, pp. 471-480.

[3] S. Sadek, A. Al-Hamadi, B. Michaelis, and U. Sayed, "An Efficient Method for Real-Time Activity Recognition," in *Soft Computing and Pattern Recognition (SoCPaR), IEEE International Conference on*, 2010, pp. 69-74.

[4] P. Natarajan and R. Nevatia, "Online, Real-time Tracking and Recognition of Human Actions," in *Motion and video Computing, 2008. WMVC 2008. IEEE Workshop on*, 2008, pp. 1-8.

[5] P. Guo, "Real time human action recognition in a long video sequence," in *Proceedings of the 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2010, pp. 248-255.

[6] S. Cheema, A. Eweiwi, C. Thurau, and C. Bauckhage, "Action recognition by learning discriminative key poses," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International*, 2011, pp. 1302-1309.

[7] A. A. Chaaraoui, P. Climent-Pérez, and F. Flórez-Revuelta, "Silhouette-based human action recognition using sequences of key poses," *Pattern Recognition Letters*, vol. 34, no. 15, pp. 1799-1807, 2013.

[8] A. A. Chaaraoui and F. Flórez-Revuelta, "A Low-Dimensional Radial Silhouette-Based Feature for Fast Human Action Recognition Fusing Multiple Views," *International Scholarly Research Notices*, vol. 2014, 2014.

[9] D. Kalhor, I. Aris, I. A. Halin, and T. Moaini, "A Fast Approach for Human Action Recognition," in *Intelligent Systems, Modelling and Simulation (ISMS), IEEE International Conference on*, 2014.

[10] J. Hernández, R. Cabido, A. S. Montemayor, and J. J. Pantrigo, "Human activity recognition based on kinematic features," *Expert Systems*, vol. 31, no. 4, pp. 345-353, 2014.

[11] I. S. P. Co., "The body and its proportions," in *Art of Drawing the Human Body*. Barcelona, Spain: Parramon, 2004, pp. 13-17.

[12] F. Rosenblatt, "The Perceptron – A Perceiving and Recognizing Automaton," Cornell Aeronautical Laboratory Report 85-460-1 , 1957.

[13] H. Chih-Wei and L. Chih-Jen, "A Comparison of Methods for Multiclass Support Vector Machines," *Neural Networks, IEEE Transaction on*, vol. 13, no. 2, pp. 415-425, Mar. 2002.

[14] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, and I. Pitas, "The i3dpost multi-view and 3d human action/interaction database," in *Visual Media Production (CVMP'09) Conference for*, 2009, pp. 159-168.

[15] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 3rd ed. Academic Press, 2006.

[16] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, 1979.

[17] G. Bradski, "The OpenCv Library," *Doctor Dobbs Journal*, vol. 25, no. 11, pp. 120-126, Nov. 2000.

[18] A. Iosifidis, A. Tefas, and I. Pitas, "View-invariant action recognition based on artificial neural networks," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 3, pp. 412-424, 2012.

[19] A. Iosifidis, A. Tefas, and I. Pitas, "Multi-view action recognition based on action volumes, fuzzy distances and cluster discriminant analysis," *Signal Processing*, vol. 93, no. 6, pp. 1445-1457, 2013.

[20] M. B. Holte, T. B. Moeslund, N. Nikolaidis, and I. Pitas, "3d human action recognition for multi-view camera systems," in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), IEEE International Conference on*, 2011, pp. 342-349.

[21] N. Gkalelis, N. Nikolaidis, and I. Pitas, "View indepedent human movement recognition from multi-view video exploiting a circular invariant posture representation," in *Multimedia and Expo. ICME 2009. IEEE International Conference on*, 2009, pp. 394-397.