Local Trinary Patterns for Human Action Recognition

Lahav Yeffet and Lior Wolf
Tel-Aviv University
wolf@cs.tau.ac.il

Abstract

We present a novel action recognition method which is based on combining the effective description properties of Local Binary Patterns with the appearance invariance and adaptability of patch matching based methods. The resulting method is extremely efficient, and thus is suitable for real-time uses of simultaneous recovery of human action of several lengths and starting points. Tested on all publicity available datasets in the literature known to us, our system repeatedly achieves state of the art performance. Lastly, we present a new benchmark that focuses on uncut motion recognition in broadcast sports video.

1. Introduction

Human action recognition from video is an area of immense importance to visual surveillance, video indexing, and several other computer-vision domains. Despite of extensive research, fueled by the ongoing advancements in object recognition, the gap between the current capabilities and the applications' needs remains large.

Indeed, action recognition is challenging due to substantial variations in the video data that are caused by varying factors which include viewpoint and scale, clothing and the subject's appearance, personal style and action length, self-occlusion, multiple video objects, and background clutter.

Beyond recognition accuracy, there are other constraints on the design of action recognition methods. Ideally for several applications, such methods would work efficiently in an online manner, and require simultaneous detection of action at several possible time scales (different action lengths) and for every possible starting point.

The proposed method has a simple flow: every pixel at every frame is encoded as a short string of ternary digits (trits) by a process which compares this frame to the previous and to the next frame. The frame is then divided into $(m \times n)$ regions and the histograms of the trinary strings are computed for each of the mn region. These histograms are accumulated every few frames and the vector which contains all concatenated histograms serves as a video descriptor for a section of the video. Multiple such vectors are concatenated to represent longer videos.

The encoding process itself is based on comparing nearby patches, in a manner inspired by the self-similarity approach [17]. For every pixel of every frame, a small patch centered at this pixel is compared to shifted patches in the previous and in the next frame. In a manner pertaining to the Local Binary Pattern approach [13], one trit of information is used to describe the relative similarity of the two patches to the patch in the central frame: the shifted patch in the previous frame is more similar to the central one, the patch in the next frame shifted by the same amount is more similar, or both are approximately comparable in their similarity.

Due to its design, our method has the following characteristics: it is extremely efficient, it can be computed in an online manner, it requires no additional blocks such as optical-flow computation, and it exclusively encodes motion while disregarding all appearance information.

2. Related work

Two popular trends can be identified in the action recognition literature, obtaining top-level performance on existing benchmarks. First, there are contributions which compute representations for nearby frames (either motion-centric or to the entire frame) [5, 8, 15, 6, 21]. Such approaches usually rely on optical-flow, on appearance, or on a combination of the two. Second, there are contributions which focus on identifying space-time interest points and on representing those local entities [4, 16, 11, 10].

The local self-similarity method [17], represents space-time interest points as vectors based on the spatial histogram of similarities between a central cuboid (space-time patch) and nearby cuboids. Excellent results are shown for the space-time template matching problem. However, it is not obvious that such an approach can excel as is on existing action recognition benchmarks, which focus on the supervised learning scenario with multiple training examples.

In this work we propose to employ the self-similarity idea within an efficient representation, which is inspired by Local Binary Patterns (LBP) [13]. Our approach has links to a variant of LBP called Center-Symmetric LBP

(CS-LBP) [7], and uses a similar structure to encode motion information based on the self-similarity idea. A previous approach which combines self-similarities and CS-LBP has been recently proposed to encode face data in static images [22], showing promising results when combined with other descriptor. Trits are also not new to the LBP world: [18] suggests the use of trits based on pixel values for overcoming difficult illumination conditions.

Given the effectiveness of LBP, it is not surprising that it has been proposed for action recognition in the past. A recent application [9] is based on slicing the space-time volume along the three axis (x, y, t), and constructing LBP histograms of the xt plane and yt plane. Another application [23] is based on computing a variant of LBP to capture local characteristics of optical flow, and then representing actions as strings of local atoms.

Our system provides no direct appearance information in the absence of motion. This characteristic is shared with optical-flow based systems (e.g., [6, 5]) or tracking based systems (e.g., [1]), and is in contrast with space-time gradient methods (e.g., [10, 4, 16]), previous LBP based action recognition systems [23, 9] and exemplar based methods (e.g., [21]). A recent trend is to combine both appearance information with optical flow-based information, which leads to improved results [11, 8, 15]. This can be done in our approach as well, and is left for future work.

3. Encoding action in the frame

The various flavors of Local Binary Patterns use short binary strings to encode simple properties of the local micro-texture around each pixel. CS-LBP [7], for example, encodes at each pixel location the sign of the gradient in four different angles.

Here we propose an LBP like descriptor which captures the effect of motion on the local structure of self-similarities. Consider a small image patch moving from left to right. During its motion it will pass through a certain image location $(x - \Delta x, y)$ at time $t - \Delta t$, and continue to location (x, y) to the right at time t. This motion is probably going to induce image similarity between a patch of appropriate dimensions centered at location $(x - \Delta x, y)$ at time $t - \Delta t$ and the patch with the image center (x, y) at time t.

By itself, the increase of image similarity caused by the motion depends on the intensities of the moving patch and the appearance of the rest of the image. It may be difficult to distinguish between similarity caused by motion and similarity caused by similar static textures, without incorporating further statistics. Here we suggest to examine the similarity between a patch centered at (x, y) at time t and the patch around $(x - \Delta x, y)$ at time $t + \Delta t$ as the background statistic. One trit is used to encode whether one of the two similarities is significantly higher than the other or whether the two similarities are approximately the same. If the previous frame patch is more similar to the central patch - a value of -1 is assigned, if the patch in the next frame is more similar - a value of $+1$ is assigned. If both similarities are within a predefined threshold from each other, a value of 0 is assigned.

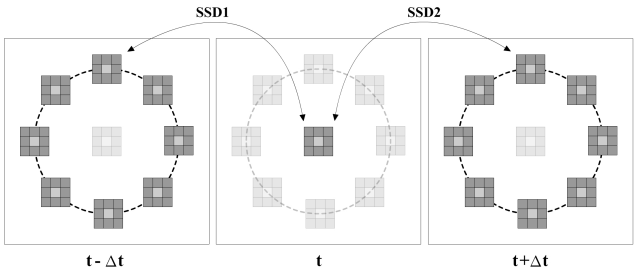


Figure 2. An illustration of the encoding process. For each of 8 different locations at time $t - \Delta t$ and the same locations at time $t + \Delta t$ SSD distances of 3×3 patches to a central patch at time t are computed. $SSD1$ and $SSD2$ are computed patch distances at one of the eight locations. One trinary bit is used to encode if $SSD1 < SSD2 - TH$ (value of -1), $|SSD1 - SSD2| < TH$ (value of 0), or $SSD2 < SSD1 - TH$ (a value of $+1$). In our system gray values are between 0 and 255, and TH is set to $1,000$. Also in our system, Δt is set to 3 frames, and the patches are spread around as close as possible using integer values to distance of 4 pixels from the center of the central patch.

Note that in the absence of significant image motion the similarities of the patch at center location (x, y) at time t to the patches at location $(x - \Delta x)$ at times $t - \Delta t$ and $t + \Delta t$ are about equal, and the value of the encoding trit is zero. This implies that no appearance information is encoded in the absence of motion.

It is worth noting that both the existence of a motion, as well as its magnitude are of importance. Consider, for example, the boxing motion in Figure. 1(a), which is taken from the KTH dataset [16]. While performing this motion the hand moves forward and the shoulders move back. This is depicted in Figure 1(b), where the direction of the motion is encoded in color. This is with contrast to the pulling motion of the boxing action, in which the opposite pattern emerges (Figure 1(c,d)).

The full 8 trit encoding is described in Figure 2. Patches at eight shifted locations at times $t - \Delta t$ and $t + \Delta t$ are compared to a central patch at time t to produce 16 similarities. In our system we use, due to its computational simplicity the SSD (sum of square differences) score as the basic distance between the patches. The lower the SSD score, the larger the similarity.

One trit is assigned for each of eight comparisons that are made between pairs of similarities that share the same spatial shifts. Thus 8 trits are used to represent each pixel in the video. Experiments done with 16 trits per pixel show marginal effect on the overall accuracy.

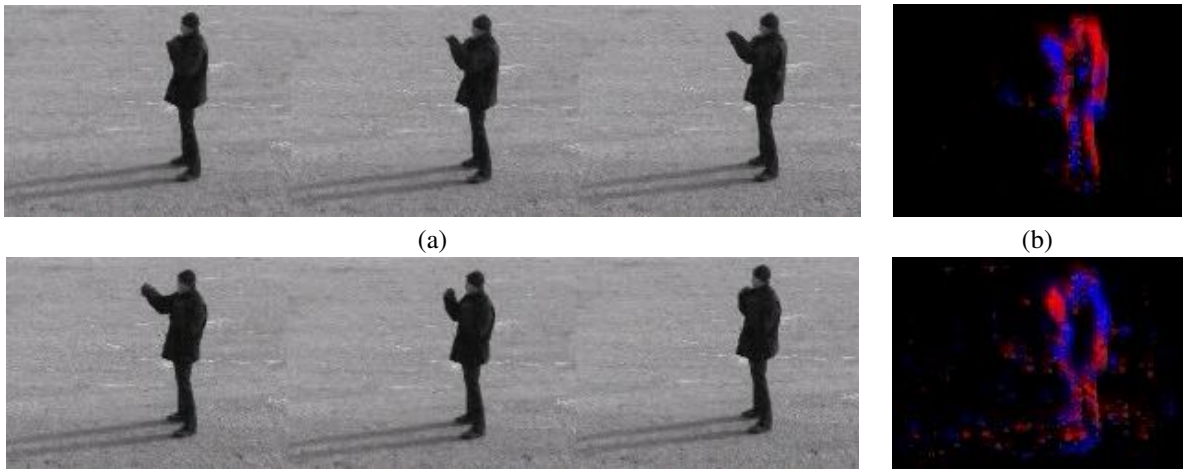


Figure 1. Two groups of nearby frames from one boxing sequence of the KTH dataset [16]. (a) Three frames from the beginning of the boxing motion. (b) One trinary digit encoding of the sequence in (a). Blue pixels indicate patches which are significantly more similar to the patch on the left in the next frame than to the patch on the left in the previous frame. Red indicates patches that are more similar to the patch of the previous frame. (c) Three frames from the end of the boxing motion, in which the hand returns. (d) The analog trit encoding of (c).

The entire frame is divided into a grid of $m \times n$ equally sized cells, and the histograms of the 8-digit trinary strings are measured in each, as described next. The string of all zeros indicate that there is no motion and is disregarded (not counted in any bin). The rest of the strings are mapped to bins in an unconventional manner, which reduces the number of possible bins from $3^8 - 1 = 6560$ to $2(2^8) = 512$, where each string is counted twice. First the positive part of the string is extracted. In this part every -1 digit is converted to the 0 digit. The positive part is then distributed between all possible binary bins. The same process repeats with the negative part which is accumulated in a separate set of 256 bins.

4. Recognizing actions

If the boundaries of the video are given, we divide the video to k equal time slices, and compute the accumulated histograms for each region among the frames of each time-slice. In practice we find it accurate enough to skip frames, and compute the histograms of no more than 10 frames per time slice. All mn region histograms for the k time slices are accumulated to one vector of length $512mnk$ which is used to represent the entire video. In order to recognize an action, we apply a classifier to those vectors. Specifically we use linear SVM, on the square-root values of the vectors. The square root operation to the values of histogram is meant to approximate the Bhattacharyya coefficients between probability distributions. This is a common practice in object recognition, e.g., [3].

A crucial question in motion recognition is the detection of the starting point and length of motion in video. In most existing benchmarks, the part of the video where the motion to be recognized reside is given. This is unrealistic for most applications, and results in an optimistic performance expectations.

We suggest to tackle the two additional unknowns (time shift and scale) by running several detectors in parallel, each observing different starting points and different scales of action length. This system can be built efficiently by reusing previous computations as depicted in Figure. 3. Since our encoding is very efficient, and since we employ straightforward linear classification, multiple detections can be achieved at better than real-time rates, our system applied to the recognition of 4 actions at 3 time scales runs at 25 frames-per-second on a modest Pentium 1.86 GH, 1 GB RAM laptop, without using the GPU or other special purpose hardware.

5. Experiments

We ran our system on all existing benchmarks we could find. We achieved state of the art performance or close to it on all datasets. In addition, we collected a new dataset for the purpose of testing performance of action recognition systems on live video where motion edges are not marked.

5.1. Parameters

There are few parameters for our system. Some, as the number and position of the patches are fixed throughout the experiments. All patches are 3×3, and are spread around the central patch at the integer approximations of a circle of radius 4. Also fixed are the value of Δt which is 3, and

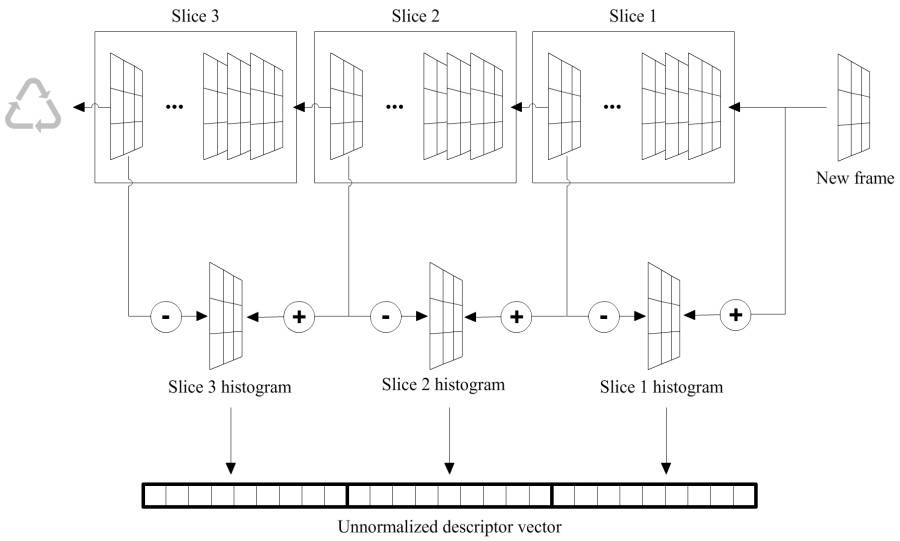


Figure 3. The structure of the online system. The next frame is encoded into histograms and is fed into the system. One accumulated histogram (containing mn sub-histograms) is kept for each time slice (three in this case). The time-slice histogram is updated upon the arrival of a new frame by adding one new frame and subtracting the frame that leaves the time-slice. The three accumulated histograms are concatenated to one representative vector, which is then (not shown) normalized so that the three parts each sum to 1, goes throw and element-wise square-root operator, and fed into the classifier. The figure depicts one detector. For multiple time scales, multiple such detectors run concurrently

the threshold on the difference between the two SSD scores, which is set to $1,000$.

The adaptations which we carry out for each dataset consist of finding the optimal grid size where (m, n) are taken to be one of $(3, 3)$,$(4, 3)$, or $(4, 4)$, the number of time slices $2 \leq k \leq 4$. The best representation out of the 9 options is selected by performing a cross validation to the examples of the first split of every experiment.

5.2. The Hollywood Human Actions Dataset

The HOHA dataset was recently collected [11] by automatically parsing movies' scripts files and automatically recognizing 8 actions. Although the labels have been manually erased, this is a challenging benchmark due to the large variability of videos labeled in the same class, and some remaining ambiguity in the true labels. The results are shown in Table 1 We compare our method (Local Trinary Patterns, LTP) with the original paper [11], which has results both for a single descriptor and for multiple descriptors combined, as well with a later on contribution [10]. Our algorithm shows preferable performance on four out of the eight categories.

Class	LTP (single)	Laptev (combined) [11]	Laptev (single) [11]	Klasser (single) [10]
Answer phone	**35.1%**	32.1%	26.7%	18.6%
Get out car	32.0%	41.5%	22.5%	22.6%
Hand shake	**33.8%**	32.3%	23.7%	11.8%
Hug person	28.3%	40.6%	34.9%	19.8%
Kiss	**57.6%**	53.3%	52.0%	47.0%
Sit down	36.2%	38.6%	37.8%	32.5%
Sit up	13.1%	18.2%	15.2%	7.0%
Stand up	**58.3%**	50.5%	45.4%	38.0%

Table 1. Average precision table for the HOHA (Hollywood human actions) dataset.

5.3. The Kissing Slapping dataset

The Kissing and Slapping dataset was collected from feature films by the authors of [14]. There are over 90 samples of the Kissing class and over 110 samples of Hitting or Slapping class. There is a large variability in the movie genres, viewpoints and type of action. Instances of action classes were annotated manually by selecting the frames corresponding to the start and end of each action, along with the spatial region of the action instance. Testing for this dataset proceeded in a leave-one-out fashion. In Table 2 we report the results obtained by our method (LTP) compared to the original results of [14]. In both categories, our system shows a higher performance than previously reported.

5.4. The UCF sports dataset

The authors of [14] have collected a large set of action clips from various broadcast sport videos. The actions in this dataset include diving, golf swinging, kicking, lifting,

Class	LTP	Rodriguez [14]
Kisses	**77.3%**	66.4%
Slaps	**84.2%**	67.2%

Table 2. Results on the Feature Films dataset (kisses and slaps)

Figure 4. Examples of the sequences collected to replace the original pole vault video clips.

horseback riding, running, skating, swinging a baseball bat, and pole vaulting. The pole vaulting sequences were removed from the original database due to copyright concerns. We therefore replaced the missing sequences with sequences we collected ourselves. Figure 4 contains example images from the new sequences.

The full dataset contains over 200 video sequences. The actions are featured in a wide range of scenes and view points. Testing for this dataset is performed using the leave-one-out framework. The confusion matrix we obtain for this set of experiments is depicted in Table 3. The overall mean accuracy we obtain for this dataset is 79.2%, compared to 69.2% reported in [14].

5.5. Older datasets

The KTH and the Weizmann datasets are well-established datasets, on which many results have been reported. Both datasets depict a relatively small person acting in front of a static background, and the datasets become somewhat easier if attention is directed to the moving parts of the scene. This is automatically given in interest-point based systems. Other systems, in order to compete on these datasets have used a filtered version of it, where the moving person is automatically extracted in a preprocessing stage [6, 8].

Here, we add an automatic attention mechanism to our method, which causes it to act in a figure-centric manner when analyzing these two datasets. The attention mechanism works by finding the image region of predefined dimensions that has the smaller number of pixels that are encoded as all zero strings. This detection is carried out efficiently by a simple application of the integral-image idea [19]. After the detection stage, the system continues in the usual manner, applied only to the detected region-of-interest.

Name	Percentile	Ref
LTP	90.1%	
LBPabs	83.8	5.5.1
LBPdir	79.6	5.5.1
Schindler	92.7%	[15]
Laptev	91.8	[11]
Jhuang	91.7%	[8]
Niebles	81.5%	[12]
Dollár	81.2%	[4]
Schüldt	71.7%	[16]

Table 4. Comparison to previous results on the KTH database

5.5.1 KTH

The KTH dataset contains sequences from six classes: walking, jogging, running, boxing and hand-waving. We adhere to the protocol of [16], where sequence from 8 people are used for training, those from another 8 for validation (we use those to tune the parameters), and the actions of 9 people are used for testing. The experiment is repeated 10 times. Table 4 reports the performance in comparison to the result reported on the leading alternative systems.

We also report results on two variants of our approach in which the self-similarities are encoded using binary digits and not trinary digits. In one system LBPabs, each of the 8 bits encodes whether the difference between the two SSD similarities of a pair of patches is larger than $1,000$. In the second system LBPdir, each bit encodes whether the patch at frame $t - \Delta t$ is similar to the central patch at time t, or whether it is the patch of the same shift at frame $t + \Delta t$ which is more similar to the one at time t.

Our results are slightly lower than the best available results on this dataset. However, it is significantly higher than both variants of encoding self-similarities using Binary strings. Table 5 shows the confusion matrix for the three local pattern systems LTP, LBPabs and LBPdir. As expected, the systems encounter the biggest difficulties in separating jogging from running.

5.5.2 Weizmann

The Weizmann action recognition dataset [2] consists of nine subjects performing nine different actions: bending down, jumping jack, jumping, jumping in place, galloping sideways, running, walking, waving one hand, and waving

	Diving	Golf Swing	Kick	Lift	Ride Horses	Run	SkateBoard	Swing	Walk	PoleVault
Diving	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Golf Swing	0.00	0.61	0.06	0.00	0.00	0.00	0.00	0.00	0.33	0.00
Kick	0.05	0.00	0.65	0.00	0.00	0.00	0.00	0.10	0.20	0.00
Lift	0.00	0.00	0.00	0.67	0.00	0.00	0.00	0.00	0.33	0.00
Ride Horses	0.17	0.00	0.08	0.00	0.67	0.00	0.00	0.00	0.08	0.00
Run	0.00	0.00	0.00	0.00	0.00	0.69	0.15	0.08	0.08	0.00
SkateBoard	0.00	0.00	0.00	0.00	0.00	0.08	0.92	0.00	0.00	0.00
Swing	0.00	0.00	0.03	0.00	0.00	0.00	0.03	0.94	0.00	0.00
Walk	0.00	0.05	0.00	0.00	0.00	0.05	0.05	0.00	0.86	0.00
PoleVault	0.00	0.00	0.00	0.00	0.04	0.04	0.00	0.00	0.00	0.92

Table 3. Confusion matrix for the Broadcast Television Action Dataset (UCF sports)

	box	clap	wave	jog	run	walk
box	0.98	0.02	0.00	0.00	0.01	0.00
clap	0.03	0.95	0.01	0.01	0.00	0.00
wave	0.02	0.02	0.96	0.00	0.00	0.00
jog	0.00	0.00	0.00	0.76	0.17	0.07
run	0.00	0.00	0.00	0.13	0.86	0.01
walk	0.01	0.00	0.00	0.08	0.01	0.90

(a)

	box	clap	wave	jog	run	walk
box	0.89	0.08	0.03	0.00	0.00	0.00
clap	0.00	0.97	0.03	0.00	0.00	0.00
wave	0.00	0.06	0.94	0.00	0.00	0.00
jog	0.00	0.00	0.00	0.78	0.00	0.22
run	0.00	0.00	0.00	0.33	0.61	0.06
walk	0.00	0.00	0.00	0.17	0.00	0.83

(b)

	box	clap	wave	jog	run	walk
box	0.89	0.08	0.03	0.00	0.00	0.00
clap	0.03	0.94	0.03	0.00	0.00	0.00
wave	0.06	0.03	0.92	0.00	0.00	0.00
jog	0.00	0.00	0.00	0.67	0.08	0.25
run	0.00	0.00	0.00	0.42	0.50	0.08
walk	0.00	0.00	0.00	0.14	0.00	0.86

(c)

Table 5. Confusion matrices on the KTH dataset for: (a) the proposed LTP algorithm, (b) the LBPabs algorithm in which each bit encodes the existence of difference in similarity above a threshold, and (c) the LBPdir algorithm in which each bit encodes the direction of the larger similarity.

both hands. The evaluation is done in a leave-one-person-out manner: 8 subjects are used for training, and the remaining one for testing. The experiment is repeated for all 9 persons, and the results are averaged.

Name	Percentile	Ref
LTP	100%	
bag-of-snip-1	100%	[15]
Blank	100%	[2]
Jhuang	98.8%	[8]
Wang	97.8%	[20]
Ali	92.6%	[1]
Dollár	86.7%	[4]
Niebles	72.8%	[12]

Table 6. Comparison to previous results on the Weizmann dataset

As Table 6 indicates, our method achieves maximal performance on this dataset. We know of two other systems with the same level of performance reported [15, 2].

5.6. the UFC database

Our system is designed to work in an online manner on live video. This design requirement stems directly from real-world applications such as surveillance video monitoring, and human-machine interface. Given the amount of video being accumulated at every moment, video indexing systems also require efficient real-time or faster processing.

Some of the previous contributions, e.g., [17] have presented results on uncut video. However, existing quantitative benchmarks are not suitable for testing online systems, since they focus on pre-cut video clips. We therefore construct a new benchmark based on UFC videos. UFC is a fighting sport, in which fights occur standing up (similar to kick-boxing), in the clinch (the fighters hold each other), or on the ground (similar to wrestling). It is therefore versatile compared to other sports and contains a myriad of different actions. In addition, some of these actions are of relevance

for surveillance applications, such as one person hitting another.

The UFC videos contain variability in view-point and in individual appearance, camera motion and shot cuts. Every action can be performed in any number of ways, and the frequency of the various actions differs considerably. In addition, in UFC videos, two fighters act at the same time, a challenge which is seldom met in previous benchmarks.

Currently, our dataset contains over 20 minutes of broadcast video. For our tests, we mark two actions that occur relatively infrequently. One is the throw/take-down action, in which one person throws another to the ground. A person can do so in any number of ways including throwing another person over the hip, or by grabbing the other person's legs. The second action is knee kick from the clinch, which is a much less versatile action. Those kicks are however harder to detect due to self-occlusion and the relatively small amount of motion involved. Figure 5 shows examples of the two actions.

Half of the video is used for training, and half of the video, which displays other fighters, is used for testing. The results are presented in Figure 6. As can be seen, for both actions, achieving a recall value significantly higher than 30% results in a sharp increase in the false positive rate. The dataset therefore poses a real challenge for our action recognition system.

6. Conclusion

We present an effective real-time system for action recognition. The system compares nearby patches and is therefore resilient to variations in texture. Moreover, it is silent in the absence of motion. Similarly to other systems, we hypothesize that incorporating appearance information would increase performance on certain benchmarks.

A new benchmark is presented for the detection of uncut action in long videos. We hope that the development of such benchmarks will be matched by further increase in performance on continuous video, and especially by the development of suitable real-time systems.

Acknowledgments

This research is supported by the Israel Science Foundation (grant No. 1214/06), the Colton Foundation, and MOST Russia-Israel Scientific Research Cooperation.

References

[1] S. Ali, A. Basharat, and M. Shah. Chaotic invariants for human action recognition. *ICCV*, Oct. 2007. 2, 6

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 2:1395–1402 Vol. 2, Oct. 2005. 5, 6

[3] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, pages 142–149 vol.2, 2000. 3

[4] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, October 2005. 1, 2, 5, 6

[5] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *IEEE International Conference on Computer Vision*, pages 726–733, Nice, France, 2003. 1, 2

[6] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. 1, 2, 5

[7] M. Heikkilä, M. Pietikäinen, and C. Schmid. Description of interest regions with center-symmetric local binary patterns. In *Computer Vision, Graphics and Image Processing, 5th Indian Conference*, pages 58–69, 2006. 2

[8] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, pages 1–8, 2007. 1, 2, 5, 6

[9] V. Kellokumpu, G. Zhao, and M. Pietikainen. Human activity recognition using a dynamic texture based method. In *BMVC*, 2008. 2

[10] A. Klaeser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008. 1, 2, 4

[11] 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*, 2008. 1, 2, 4, 5

[12] J. Niebles and F. Li. A hierarchical model of shape and appearance for human action classification. In *CVPR*, 2007. 5, 6

[13] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 24(7):971–987, 2002. 1

[14] M. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008. 4, 5

[15] K. Schindler and L. van Gool. Action snippets: How many frames does human action recognition require? *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. 1, 2, 5, 6

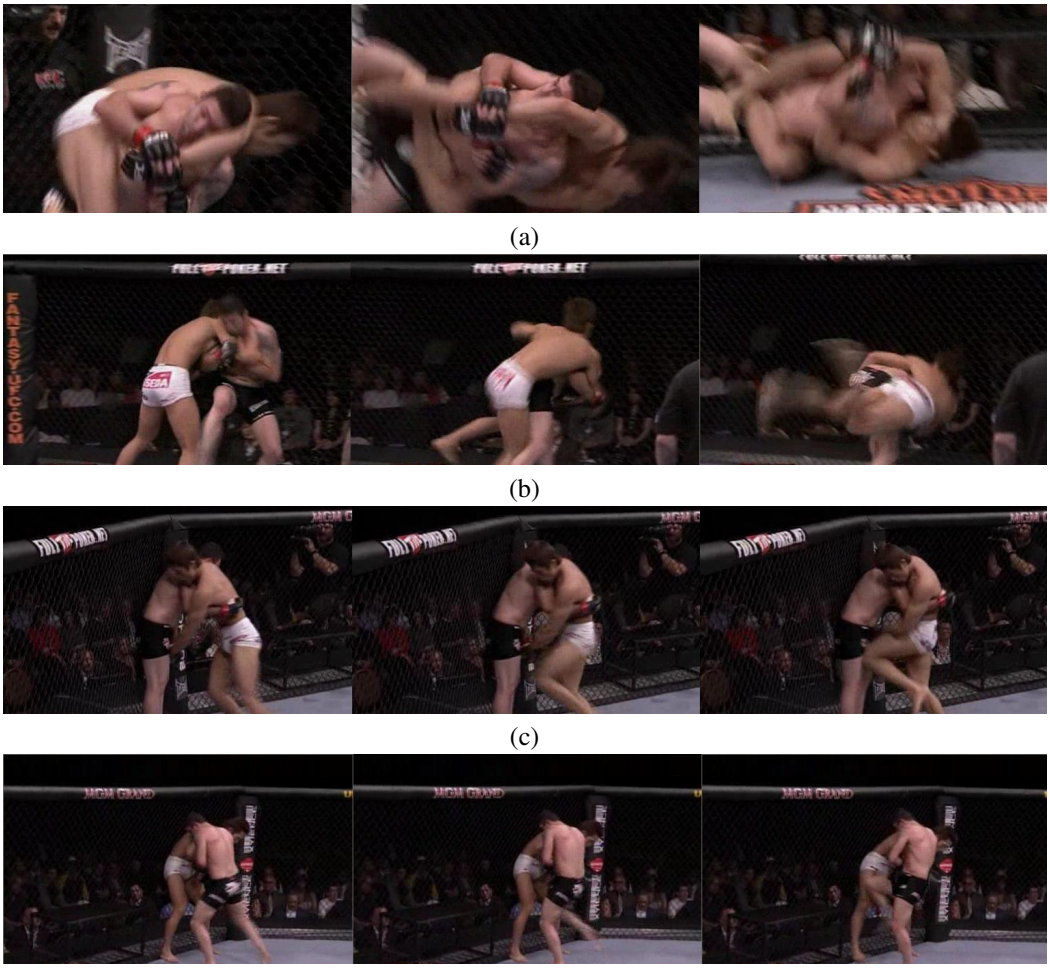
[16] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, 2004. 1, 2, 3, 5

[17] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *CVPR*, June 2007. 1, 6

[18] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. In *International Workshop on Analysis and Modeling of Faces and Gestures*, 2007. 2

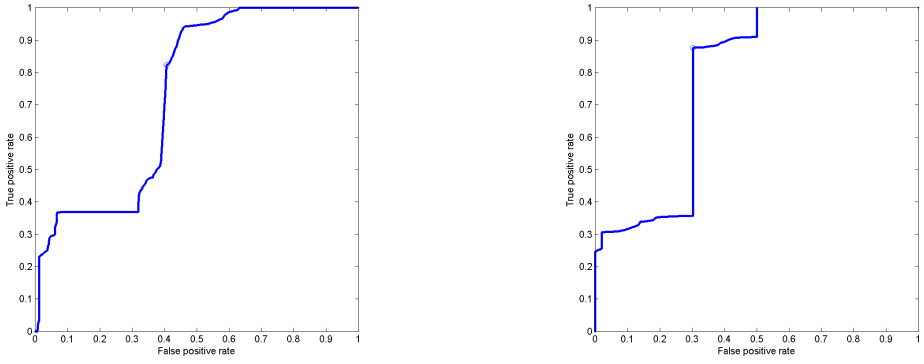
[19] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision - to appear*, 2002. 5

[20] L. Wang and D. Suter. Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. In *CVPR*, 2007. 6



(a) (b) (c) (d)

Figure 5. Example actions from the UFC database. (a),(b) Examples of the Throwing action. (c),(d) Examples for the action of Knee-kick from the clinch.



(a) (b)

Figure 6. ROC curves for the UFC dataset. (a) for the Throw action. (b) for the Knee-kick action.

[21] D. Weinland and E. Boyer. Action recognition using exemplar-based embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage*, pages 1–7, 2008. 1, 2

[22] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Faces in Real-Life Images ECCV work-*

shop, 2008. 2

[23] C. Yang, Y. Guo, H. S. Sawhney, and R. Kumar. Learning actions using robust string kernels. In A. M. Elgammal, B. Rosenhahn, and R. Klette, editors, *Workshop on Human Motion*, volume 4814 of *Lecture Notes in Computer Science*, pages 313–327. Springer, 2007. 2