Violent Flows: Real-Time Detection of Violent Crowd Behavior*

Tal Hassner
Open University of Israel
hassner@openu.ac.il

Yossi Itcher
Open University of Israel
Yossi.itcher@gmail.com

Orit Kliper-Gross
Weizmann Institute of Science
orit.kliper@weizmann.ac.il

Abstract

Although surveillance video cameras are now widely used, their effectiveness is questionable. Here, we focus on the challenging task of monitoring crowded events for outbreaks of violence. Such scenes require a human surveyor to monitor multiple video screens, presenting crowds of people in a constantly changing sea of activity, and to identify signs of breaking violence early enough to alert help. With this in mind, we propose the following contributions: (1) We describe a novel approach to real-time detection of breaking violence in crowded scenes. Our method considers statistics of how flow-vector magnitudes change over time. These statistics, collected for short frame sequences, are represented using the VIolent Flows (ViF) descriptor. ViF descriptors are then classified as either violent or non-violent using linear SVM. (2) We present a unique data set of real-world surveillance videos, along with standard benchmarks designed to test both violent/non-violent classification, as well as real-time detection accuracy. Finally, (3) we provide empirical tests, comparing our method to state-of-the-art techniques, and demonstrating its effectiveness.

1. Introduction

There is no question that video surveillance equipment can be easily and cheaply deployed to monitor practically any environment. The value of doing so, however, is indeed questioned [2]. Surveillance systems are often ineffective due to insufficient numbers of trained supervisors watching the footage and the natural limits of human attention capabilities [14]. This is understandable, when considering the huge numbers of cameras that require supervision, the monotonic nature of the footage, and the alertness required to pick up on events and provide an immediate response. In fact, even the seemingly simpler task of searching recorded videos, off-line, for events that are known to have happened, requires the aid of Computer Vision systems for video retrieval (e.g., [26]) and summarization [27].

Here, we focus on the task of detecting outbreaks of crowd violence, as it happens, from surveillance video cameras. Such videos typically do not have audio tracks, and, of course, subtitles and other contextual sources of information are non-existent. The footage is often far below motion picture quality, and so color cues are not reliable and neither are the details required for fine-scale action recognition. Some action recognition techniques are designed to analyze a single dominant action in the video. Here, however, videos present crowds, and we do not know a-priori who will participate in the violence. Finally, crowd scenes are especially challenging as they present constant, often monotonous, spatially unconstrained, human motion. This may not only reduce the effectiveness of a human observing the videos over long periods of time, but it can also flood a Computer Vision system with large quantities of motion information, making methods relying on interest points too time consuming. Figure 1 illustrates the type of scenarios we consider here by providing some examples from our database of both violent and non-violent crowd behavior.

Figure 1. Examples of violent (bottom-left) and non-violent (top-right) crowd behavior in "real-world" videos.

In order to design a system capable of operating in real-time, we forgo high-level shape and motion analysis (e.g., [1]) and intensive processing [15], instead following the example of methods for dynamic texture recognition, such as [13], in collecting statistics of densely sampled, low-level features. For the purpose of violence detection in crowded scenes we show that accuracy can be achieved, without compromising processing speed, by considering how flow-vector magnitudes change through time. We collect this information, over short frame sequences, in a representation which we call the VIolent Flows (ViF) de-

*Data available from: www.openu.ac.il/home/hassner/data/violentflows/

scriptor (Sec. 3.1). ViF descriptors are then efficiently labeled as violent or non-violent using a standard linear Support Vector Machine (SVM).

In order to test the accuracy of our method we require suitable data and benchmarks. Few video collections are available for testing violence detection performance, and none that we are aware of focus on the problem described here. We have therefore assembled our own collection of videos, presenting both violent and non-violent crowd behaviors. Our videos were all downloaded from the web and therefore represent unconstrained, “in-the-wild” conditions and scenes. We tested both our own method, as well as existing state-of-the-art techniques on violence classification and violence detection benchmarks designed using this collection. Our tests clearly demonstrate the wide performance margin, in favor of the method proposed here.

2. Previous work

Action recognition. Violence detection, is a particular problem within the greater problem of action recognition. Methods for action recognition can roughly be classified as either local, interest-point based approaches, or global, frame-based methods. Interest-point based methods begin by first detecting space-time key-points [9, 24]. Descriptive information is then extracted at each of these points using one of several space-time descriptors (see for example: [11, 17, 19]). A video can then be represented using, e.g., Bag-of-Feature techniques (as in [19, 23]). These methods are often very resilient to camera motion and have been shown to provide excellent performance on a number of challenging benchmarks [16, 19, 23], however, when videos contain too few space-time interest points (e.g., little motion) or too much motion (as in our scenarios), they may fail to efficiently provide meaningful representations.

The alternative of considering whole frames, or frame parts, often builds on dense flow estimation between successive frames [3, 4] or high-level appearance models [1]. Related to crowd videos are the methods of [12] and more recently Rodriguez et al. [28]. Both these methods are data-driven and require matching parts of the query video – frame segments in [12] and spatiol-temporal cubes in [28] – to exemplars in a pre-collected database. Searching the database for matching exemplars would be impractical for the applications considered here.

Violence detection. Often, “violence detection” refers to detecting violent scenes in motion pictures and TV broadcasts. The term “violence” may refer to anything from explosions to more subtle actions. In such cases, audio may provide important additional information for detection [8, 20]. Sometimes a significant change in the scene (a “surprising event”) may be considered an act of violence. Boiman and Irani proposed an approach for detecting unexpected events in videos by using a data-driven approach [6]. It is not straightforward to apply their method for real-time processing. Hendel et al. [10], on the other hand, describe a more efficient, probabilistic technique. Their method, however, assumes that the scene can be characterized using multiple space-time tubes, each containing an object moving in the scene. This requirement is often impractical in videos of crowds.

Dynamic textures. Videos of crowds may be described as produced by a stochastic process, stationary in both space and time. Such videos are often referred to as dynamic textures [13]. Although the videos we focus on here are not necessarily stationary – different parts of the frames may have different motion patterns – it is reasonable to consider analyzing them using dynamic texture recognition techniques. Indeed, over the past decade, such methods have been successfully applied to varying scenes, from pure textures to facial expression recognition. Recently, Local Binary Patterns (LBP), originally proposed for face and texture recognition in 2D images [25] and extended for 3D videos, have proven both effective and efficient in recognizing motion patterns [13, 33]. Inspired by these methods, the Local Trinary Patterns (LTP) of [32] has demonstrated state-of-the-art performance on action recognition tasks.

Benchmarks for action recognition. Video benchmarks have recently shifted focus, presenting more and more videos obtained “in-the-wild”, typically downloaded from online repositories such as YouTube. For a recent, comprehensive survey of such benchmarks, see [16]. Few data sets, however, provide surveillance footage and none provide surveillance footage capturing violent crowd behavior. Although some test sets have been assembled for the purpose of violence detection, these typically focus on violence occurring between two (or very few) people [5] or contain high quality motion picture and TV footage (e.g., the “slaps and kisses” data-set [29]). The videos assembled here, described in Sec. 4, present challenging, real-world, crowd scenes. We design both a straightforward, five-fold, cross validation test for violence classification accuracy, as well as tests for violent action detection.

3. Violence in crowded scenes

We make the following assumptions on the footage and the problem at hand: (1) Viewpoints are far from the scene, and therefore capture many people appearing in low resolution. (2) Processing must be kept at real-time; frame processing should require less than 1/25 seconds per frame on a standard computer and a detection should be made within a few seconds of the outbreak of violence.

Given a video sequence $\mathcal{S}$ of frames $\{f_1, f_2, \ldots\}$ we consider two related but different tasks. The first is *violence classification*: The video $\mathcal{S}$ is assumed to be segmented temporally, containing T frames portraying either violent or non-violent crowd behavior. The goal is to classify $\mathcal{S}$ accordingly. The second is *violence detection*: Here, we assume an input stream of frames and the goal is to detect the change from non-violent to violent behavior, with the shortest delay from the time (frame) that the change occurred. Moreover, as mentioned above, this goal must be achieved with processing performed faster than frame-rate.

Existing work [30] has shown that under certain circumstances, less than ten video frames are required for reliable action classification. We consider such sub-second delays acceptable for a detection system and so reduce the second problem to the first by processing short frame sequences separately, classifying each one as either violent or non-violent; a detection is reported once a violent sub-sequence of frames is thus encountered. We next describe how each frame sequence is represented and classified.

3.1. The ViF representation

Given a sequence of frames, $\mathcal{S}$, we produce the VIolence Flows (ViF) descriptor by first estimating the optical flow between pairs of consecutive frames. This provides for each pixel $p_{x,y,t}$, where t is the frame index, a flow vector $(u_{x,y,t}, v_{x,y,t})$, matching it to a pixel in the next frame $t+1$. Here, we consider only the magnitudes of these vectors: $m_{x,y,t} = \sqrt{(u^2_{x,y,t} + v^2_{x,y,t})}$. Doing so is in some sense a throwback to some early action recognition techniques which also relied on flow-vector magnitudes for processing actions [21]. There are some important differences, however, between those earlier approaches and our own.

Unlike previous methods, we do not consider the magnitudes themselves, but rather how they *change* over time. Our rationale is that although flow vectors encode meaningful temporal information, their magnitudes are arbitrary quantities: they depend on frame resolution, different motions in different spatio-temporal locations, etc. By comparing magnitudes we obtain meaningful measures of the significance of observed motion magnitudes in each frame compared to its predecessor. This is somewhat related to the self-similarity descriptor of [31] and its extension to action recognition using the LTP descriptors [32]. Unlike them, however, we consider similarities of *flow-magnitudes in time*, rather than local appearances.

Specifically, for each pixel in each frame we obtain a binary indicator $b_{x,y,t}$, reflecting the significance of the change of magnitude between frames:

$$b_{x,y,t} = \begin{cases} 1 & \text{if} \quad |m_{x,y,t} - m_{x,y,t-1}| \geq \theta \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Where θ is a threshold adaptively set in each frame to the average value of $|m_{x,y,t} - m_{x,y,t-1}|$. Doing so provides us with a binary, magnitude-change, significance map b_t for each frame f_t. We next compute a mean magnitude-change map by simply averaging these binary values, for each pixel, over all the frames $f_t \in \mathcal{S}$:

$$\bar{b}_{x,y} = \frac{1}{T} \sum_t b_{x,y,t}. \tag{2}$$

In its simplest form, the ViF descriptor is a vector of frequencies of quantized values $\bar{b}_{x,y}$. If the crowd motion patterns were indeed spatially stationary, this may suffice. In practice, however, we found that different spatial regions have different characteristic behaviors. The ViF descriptor is therefore produced by partitioning $\bar{b}$ into $M \times N$ non-overlapping cells and collecting magnitude change frequencies in each cell separately. The distribution of magnitude changes in each such cell is represented by a fixed-size histogram. These histograms are then concatenated into a single descriptor vector.

3.2. Classification with ViF descriptors

We use the ViF descriptors for classification in two distinct manners: (1) As global descriptors, extracted for a frame sequence as a whole or (2) as proxies used to produce a Bag-of-Features representation for each sequence.

Global descriptors. For a given sequence $\mathcal{S}$ we produce its ViF representation. Each such vector is then classified as representing an either violent or non-violent video. In practice, we found the ViF representation to capture meaningful, descriptive information, thus providing high classification scores even using simple linear support vector machines (SVM) [7] as the underlying classifier. As a consequence, real-time violence detection is achieved by considering short frame sequences, encoding each using its ViF descriptor and then immediately classifying it.

Bag-of-Features. Although ViF descriptors were designed with crowd behavior videos in mind, it is natural to consider how well they perform on "non-textured" actions and general action recognition tasks. In Section 5.2 we present such results using the ASLAN benchmark [16]. On large enough training sets, we take the frequency vectors produced for each cell as local video descriptors. These are analogous to the descriptors produced by using existing STIP techniques. Here, however, we produce our own descriptors in a uniform, $M \times N$ grid. These descriptors are then quantized into a visual vocabulary using k-means. A whole video sequence is then represented using the frequencies of the ViF words it includes. The bags of words are then used according to the application at hand (Section 5.2).

4. The violent crowds data-set and benchmarks

Although data-sets which include videos for action recognition are by no means rare, we know of none suit-

Table 1. **Violence/Non Violence Database Statistics**

General statistics:	
♯ of videos	246
♯ unique urls	214
♯ unique YouTube titles	218
Video statistics:	
Shortest video duration	1.04 sec.
Longest video duration	6.52 sec.
Average video duration	3.60 sec.

able for testing violent crowd behavior. We therefore assembled our own database of videos for use in both violence classification and violence detection tasks. To avoid introducing biases for particular scenes or behaviors, and at the same time provide a wide range of challenging real-world viewing conditions, our data is collected from YouTube. It therefore includes videos produced under uncontrolled, in-the-wild conditions, presenting a wide range of scene types, video qualities and surveillance scenarios. Table 1 provides additional statistical information on our database. The movies themselves are all de-interlaced and stored as AVI files. All the videos were compressed using the DivX codec (mpeg4), with the frames resized to 320×240 pixels.

4.1. Benchmark protocols

We design two separate benchmarks on our video set.

Classification. The first benchmark is a five-fold cross-validation, classification test. We split the video set into five sets: half the videos in each set portray violent crowd behavior and half non-violent behavior. In some cases, different videos originated from the same YouTube clip or the same scene. In such cases, these videos are all included in the same set (the sets were mutually scene-exclusive).

The classification tests use a five-fold cross validation test. Five tests are performed; in each test, four of sets are used for training (including SVM training and vocabulary generation, when required). Violence labeling is then performed on the remaining set. Results are reported as both mean prediction accuracy (ACC) $\pm$ standard deviation (SD) as well as the area under the ROC curve (AUC).

Detection. To evaluate the accuracy and reaction-time of a violence detection method, we consider videos beginning with non-violent behavior which turns to violence mid-way through the video. 21 such videos exist in our collection. We manually mark the frame in each video where this transition happens. The goal is to detect the violence as close to its manually specified outbreak as possible. Methods are required to process the videos, with frames provided sequentially. We require results on this test to present, in a graph, the percent of violence detections (percent of videos where violence was correctly detected) for increasing delays in time from violence outbreak. Different methods can then be compared by their accuracy vs. the time they require to detect the violence. Here, all training is performed on the videos which were *not* included in the detection set.

Table 2. Classification results on our crowd violence database, mean over 5-folds cross validation. We report mean accuracy ($\pm$ Standard Deviation) and AUC.

Method	Accuracy ($\pm$ SD)	AUC
LTP [32]	71.53 $\pm$ 0.17 %	79.86
HOG [19]	57.43 $\pm$ 0.37 %	61.82
HOF [19]	58.53 $\pm$ 0.32 %	57.60
HNF [19]	56.52 $\pm$ 0.33 %	59.94
ViF	81.30 $\pm$ 0.21 %	85.00

5. Experiments

Our method was implemented in MATLAB, the optical-flow code available from [22], and linear SVM [7]. We have made few attempts to optimize the few parameters of our method, and so improved performance may be obtained by exploring other values. Here, we report the values used throughout our tests: We use a grid size of $M \times N = 4 \times 4$. For the violence/non-violence classification task we consider the whole video at once, i.e. Equation 2 averages over all the frames in the video to produce a single ViF descriptor. We use 20-bin histograms no matter the number of frames in the video. For real-time detection, Equation 2 averages frames in five-frame temporal windows, classifying each one separately and appropriately using six-bin histograms. We further found that it is enough to process one in every three frames for accurate temporal detection.

We compare our method to existing state-of-the-art techniques, representing two different approaches to action recognition. The first is the interest-point driven method of [18] as used in [16]. We use the implementation of [19] and test all three spatio-temporal descriptors it provides: HOG, HOF, and HNF. We use the videos included in the training set to produce a vocabulary of $6,000$ visual words using k-mean. Each video is then represented using a single frequency vector of size $6,000$ L1 normalized.

The second method we compare with is the LTP descriptor of [32]. LTP, like ViF, is a frame-based descriptor. We therefore report its performance using the same pipeline used for our ViF descriptor.

5.1. Crowd violence database tests

We begin by presenting ViF performance on the database and benchmarks we have assembled for the purpose of violence classification and detection in crowds. Table 2 presents classification results on our five-fold cross validation test as described in Section 4.1. The ViF representation far outperforms the other methods tested. Unsurprisingly, the STIP representations, better suited for "structured

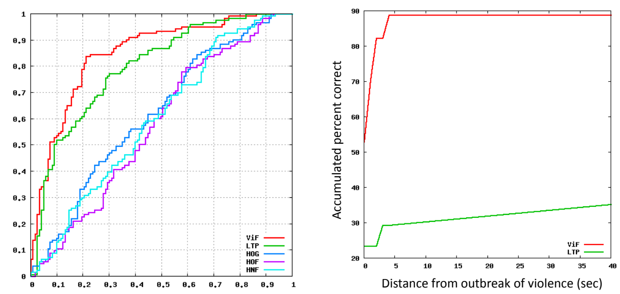

Figure 2. **Left:** Violence classification ROC curves for the various methods, averaged over 5-folds of our benchmark. **Right:** Real-Time detection results: ViF detects more violent scenes than [32] and does so sooner to the violence outbreak

Table 3. Detection results on our benchmark (see text for details).

Method	LTP [32]	ViF
Success	35.29%	88.23%
Processing time per frame (ms)	10	30
Relative success by time to detection:		
1 Frame	23.53%	52.94%
1 Sec	0%	17.65%
10 Sec	5.88%	0%

videos", rather than the more textural videos in our data set, performed at almost chance. ROC curves of all tested methods are provided in Figure 2 (Left).

Our real-time detection tests were performed on a 3Gb RAM, Intel core i7 computer running Windows Vista. These results are presented in Figure 2 (Right). We compare only ViF to LTP; STIP approaches performed too slowly for real-time processing, requiring, 0.28 seconds per-frame just for STIP feature extraction. Evidently, ViF detected far more violent scenes correctly, compared to LTP. It was furthermore far faster to detect the violence, typically in less than a second from its outbreak. Table 3 summarizes these scores, providing also run-times for the two methods. Both operated at faster than frame-rate on our computer, ViF requiring more time to compute optical-flow.

5.2. Non-crowd behavior tests

Although ViF was designed with for crowd violence detection, it is natural to ask: how well ViF performs in action classification tasks of "non-textured" video scenes? Here we report the performance of the ViF descriptor on two such action classification benchmarks.

Hockey violence classification. The Hockey data set [5] was presented for testing methods designed to classify videos as violent or non-violent between two (or a few) participants. The set contains $1,000$ clips divided into five splits, each containing 100 violent and 100 non-violent scenes. Methods are required to detect violence in a 5-folds cross validation test. Existing results on this set were obtained using STIP descriptors, representing each video using a Bag-of-Features. Table 4 shows our own result, the one we obtained with LTP, and the state-of-the-art performances [5] with STIP [18, 19]. ViF obtains performance comparable to using small STIP vocabularies. With larger vocabularies, STIP outperform ViF. This improved performance comes at a computational price, making such methods impractical for real-time processing.

Table 4. Classification results of various methods on the Hockey set of [5], averaged over five-folds cross validation scheme. All the STIP results are as reported in [5].

Method	Accuracy $\pm$ SE
STIP(HOG) Vac50 [5]	87.8%
STIP(HOF) Vac50 [5]	83.5%
moSTIP Vac50 [5]	87.5%
STIP(HOG) Vac1000 [5]	91.7%
LTP [32]	71.90 $\pm$ 0.49 %
ViF	82.90 $\pm$ 0.14 %

The ASLAN benchmark. To our knowledge, the Action Similarity Labeling Challenge (ASLAN) data set [16] is the most recent and comprehensive data set for testing action recognition methods. It includes thousands of videos portraying hundreds of human-performed actions. The goal of its accompanying benchmark is to decide if two videos present actors performing the same action, or not ("same" / "not-same" classification). Due to the un-textured nature of all the videos in the ASLAN set, and the high variability of the actions included in this set, ASLAN is highly *un*-suitable for the ViF descriptor. Nevertheless, the results reported in Table 5 demonstrate that ViF performance is comparable to other single-descriptor based methods reported in [16], while being far faster to extract. In these tests, we used the Bag-of-Words representation with ViF, as described in Sec. 3.2.

6. Conclusions

Timely detection of violent outbreaks in crowds may mean the difference between life and death. Despite the significance of this task, it has received little attention in the past. Here, we make several important contributions towards the design of a system for detecting such events: We describe a novel means for efficient crowd violence detection. To test our system, as well as existing and future methods, we assemble a challenging data-set of related videos along with standard benchmarks. Finally, we demonstrate performance of both our own technique as well as existing ones on our own benchmarks and other video benchmarks.

Interestingly, our ViF outperforms existing techniques by relying on magnitudes of the optical-flow fields alone. Although action recognition techniques have in the past been designed based on flow field magnitudes, more elab-

Table 5. Same/not-Same classification results of ViF and STIP the ASLAN video collection of [16], averaged over 10-folds cross validation scheme. All the STIP results are as reported in [16].

Method	Accuracy $\pm$ SE	AUC
HOG [16]	59.82 $\pm$ 0.82%	63.2%
HOF [16]	56.68 $\pm$ 0.56%	58.9%
HNF [16]	59.47 $\pm$ 0.66%	63.3%
ViF	56.57 $\pm$ 0.25%	58.2%

orate methods have since evolved, utilizing additional sources of information. Here we show that when considered within a suitable frame of reference – by comparing their values from one frame to the next – coupled with spatial pooling, an accurate, computationally efficient representation emerges. We show that this representation is particularly potent when applied to the problem of detecting abnormal, specifically violent, crowd behavior.

References

[1] M. Abdelkader, W. Abd-Almageed, A. Srivastava, and R. Chellappa. Silhouette-based gesture and action recognition via modeling trajectories on riemannian shape manifolds. *CVIU*, 115(3):439–455, 2011.

[2] R. Akers and C. Sellers. *Criminological theories: introduction, evaluation, and application*. Oxford University Press, 2008.

[3] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *TPAMI*, 32(2):288–303, 2010.

[4] R. Basri, T. Hassner, and L. Zelnik-Manor. Approximate nearest subspace search. *TPAMI*, 33(2):266–278, 2011.

[5] E. Bermejo Nievas, O. Deniz Suarez, G. Bueno García, and R. Sukthankar. Violence detection in video using computer vision techniques. In *CAIP*, pages 332–339, 2011.

[6] O. Boiman and M. Irani. Detecting irregularities in images and in video. *IJCV*, 74(1):17–31, 2007.

[7] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *TIST*, 2(3):27:1–27:27, 2011. www.csie.ntu.edu.tw/∼cjlin/libsvm.

[8] M. Cristani, M. Bicego, and V. Murino. Audio-visual event recognition in surveillance video sequences. *IEEE Trans. Multimedia*, 9(2):257–267, 2007.

[9] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, pages 65–72, 2005.

[10] A. Hendel, D. Weinshall, and S. Peleg. Identifying surprising events in videos using bayesian topic models. *ACCV*, pages 448–459, 2011.

[11] M. Kaâniche and F. Brémond. Recognizing gestures by learning local motion signatures of HOG descriptors. *TPAMI*, 2012.

[12] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *ICCV*, pages 1–8, 2007.

[13] V. Kellokumpu, G. Zhao, and M. Pietikainen. Human activity recognition using a dynamic texture based method. In *BMVC*, pages 1–10, 2008.

[14] H. Keval. *Effective, design, configuration, and use of digital CCTV*. PhD thesis, University College London, 2009.

[15] O. Kliper-Gross, T. Hassner, and L. Wolf. One shot similarity metric learning for action recognition. *Similarity-Based Pattern Recognition*, pages 31–45, 2011.

[16] O. Kliper-Gross, T. Hassner, and L. Wolf. The action similarity labeling challenge. *TPAMI*, (99):1–1, 2012.

[17] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *CVPR*, pages 2046–2053, 2010.

[18] I. Laptev. On space-time interest points. *IJCV*, 64(2):107–123, 2005.

[19] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8, 2008.

[20] J. Lin and W. Wang. Weakly-supervised violence detection in movies with audio and video based co-training. *Advances in Multimedia Information Processing-PCM*, pages 930–935, 2009.

[21] J. Little and J. Boyd. Recognizing people by their gait: the shape of motion. *J. of Comp. Vision Research*, 1(2):1–32, 1998.

[22] C. Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis.* PhD thesis, Massachusetts Institute of Technology, May 2009. people.csail.mit.edu/celiu/OpticalFlow/.

[23] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos 'in the wild'. In *CVPR*, pages 1996–2003, 2009.

[24] J. C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *CVPR*, pages 1–8, 2007.

[25] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*, 24(7), 2002.

[26] N. Petrovic, N. Jojic, and T. Huang. Adaptive video fast forward. *Multimedia Tools and Applications*, 26(3):327–344, 2005.

[27] Y. Pritch, S. Ratovitch, A. Hendel, and S. Peleg. Clustered synopsis of surveillance video. In *Advanced Video and Signal Based Surveillance*, pages 195–200, 2009.

[28] M. Rodriguez, S. Ali, and T. Kanade. Tracking in unstructured crowded scenes. In *ICCV*, pages 1389–1396, 2009.

[29] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach: A spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, pages 1–8, 2008.

[30] K. Schindler and L. V. Gool. Action snippets: How many frames does human action recognition require? In *CVPR*, pages 1–8, 2008.

[31] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *CVPR*, pages 1–8, 2007.

[32] L. Yeffet and L. Wolf. Local trinary patterns for human action recognition. In *ICCV*, pages 492–497, 2009.

[33] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. In *TPAMI*, volume 29, pages 915–928, 2007.