

# Detecting Interesting Events Using Unsupervised Density Ratio Estimation

Yuichi Ito<sup>1,\*</sup>, Kris M. Kitani<sup>2</sup>, James A. Bagnell<sup>2</sup>, and Martial Hebert<sup>2</sup>

<sup>1</sup> Nikon Corporation, Shinagawa, Tokyo 140-8601 Japan

<sup>2</sup> Carnegie Mellon University, Pittsburgh, PA 15213 USA

**Abstract.** Generating meaningful digests of videos by extracting interesting frames remains a difficult task. In this paper, we define interesting events as unusual events which occur rarely in the entire video and we propose a novel interesting event summarization framework based on the technique of *density ratio estimation* recently introduced in machine learning. Our proposed framework is unsupervised and it can be applied to general video sources, including videos from moving cameras. We evaluated the proposed approach on a publicly available dataset in the context of anomalous crowd behavior and with a challenging personal video dataset. We demonstrated competitive performance both in accuracy relative to human annotation and computation time.

**Keywords:** Video Summarization, Density Ratio Estimation.

## 1 Introduction

While the amount of video data from personal cameras has been increasing exponentially, the raw content of any long video is often uninformative and only a small portion of the video contains interesting information. A framework that could automatically detect and highlight interesting events within a video would significantly improve the efficiency of video analysis by focusing attention on the most salient content. While it would be impossible to anticipate the interests of the viewer without extensive training data, at least being able to filter out frames of common or uninteresting events would be very valuable. In fact, commercial products, such as Magisto [1] were introduced to address this problem.

We explore an event summarization framework based on an unsupervised classification technique to select frames (Figure 1). We assume that the input video can be described by a nominal distribution of frames, described by visual features, plus a fraction of outlier frames which do not fit the nominal distribution. In that model, the “interesting” frames selected by the algorithm correspond to unusual events which occur rarely in the entire video. This task is particularly well suited to unedited consumer videos which often include large segments of repeating or uninformative material. Importantly, the approach is unsupervised so that the level of interest of a frame is defined relative to the input video rather than relative to some fixed training set which may have little relation with the input video.

---

\* This work was done while the author was at Carnegie Mellon University.

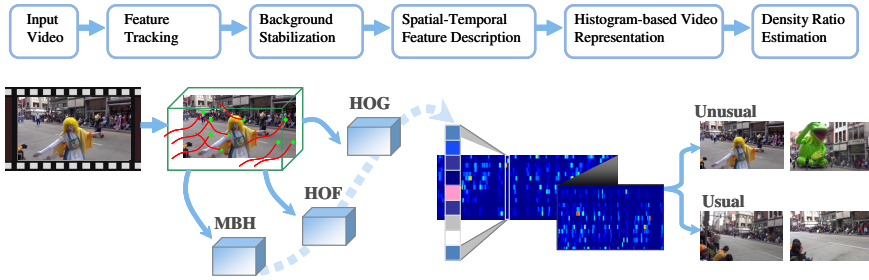


Fig. 1. Overview of our proposed framework

We encode each frame of an input video by a set of quantized spatio-temporal feature descriptors while eliminating the noise due to background motion. This approach is well suited to detecting spatio-temporal salient events. e.g. salient actions, scene changes, etc. We divide the entire set of features from the input video into two sections, and we train a logistic classifier on the corresponding two-class problem, following discriminative density ratio techniques introduced in machine learning [2]. In reality, we use different splits of the video and combine the outputs of the corresponding classifiers to get a combined detection score.

## 1.1 Related Work

Our task is to detect interesting events from general video sources, which is related to two broad areas of research: video summarization and anomaly detection. There are various event detection approaches in complex visual scenes [3,4]. Most of the approaches for video summarization are based on video skimming, which is a technique to select short video clips. Previous work on video skimming [5] can be classified into two categories: summary oriented and highlight oriented. Summary oriented methods keep the essential part of each shot and generates brief summaries [6,7]. In contrast, the highlight oriented methods only select a few interesting and salient parts of the original video. Movie trailers and highlights of sports events are examples of this type [8]. The latter methods are most closely related to our task. However, defining which video shots to be highlighted is a subjective and difficult process. Detecting unusual events, or “anomalies”, is also a key component of video surveillance systems. Although the details vary depending on the specific application, anomaly detection generally involves detecting events which occur rarely using model or saliency based [9,10,11], sparse coding [12], trajectory analysis [13], or HMM models [14].

## 2 Proposed Method

### 2.1 Density Ratio Estimation

For the sake of explanation, let us first consider a slightly different problem in which we have two separate videos. One video, called the “reference” or  $R$ ,

does not contain any interesting events. The second video called the “input” video or  $I$ , is the one in which we wish to find the interesting events, i.e., the ones that are sufficiently different from  $R$ . We also assume that each frame of both videos is represented by a feature vector  $f$ . In this setting, the task is to decide whether each frame of  $I$ ,  $f_I$  is unusual, i.e., sufficiently different from the other frames in the video. One natural approach [15] is to model the probability density of the frame features from the reference video  $P(f|R)$ . One can then classify those frames  $f_I$  from  $I$  for which  $P(f|R)$  is low as interesting or unusual events. This density estimation approach has several major issues. First, density estimation in a high dimensional space is generally a difficult problem and may be, in fact, unnecessary to detect anomalies. In addition, because it is based on the likelihood of feature occurrence in the video, the approach cannot account for the prior frequency of occurrence of any feature value in *any* video.

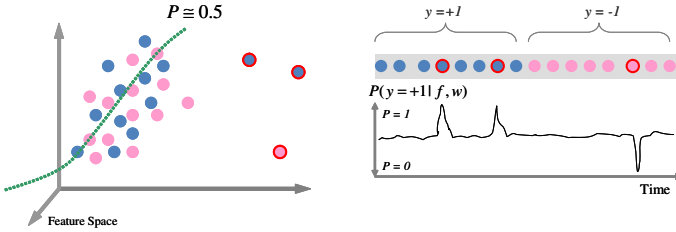
The alternate approach that we explored is known as density ratio estimation [2]. This approach exploits the insight from machine learning that it is much easier to learn a ratio of two probability densities in a high dimensional space than to learn each separately. This is why density ratio estimation is used in many fields, such as outlier detection [2] and change points detection [16], etc. In this model, we view  $R$  and  $I$  as training data for a two-class classification problem in which we assign a label  $y = +1$  if the frame is classified as originating from  $R$  and  $y = -1$  if it is classified as originating from  $I$ . Under this classification task definition, we can estimate the density ratio  $\rho(f) = \frac{P(y=+1|f)}{P(y=-1|f)}$  from all the frames in  $R$  and  $I$ . For a given frame  $f_I$  from  $I$ ,  $\rho(f_I)$  should be close to 0.5 or greater if the frame is not part of an anomalous segment because, by definition, non-anomalous features are similarly distributed between reference and input videos, whereas the probability is close to 0 if the feature comes from an anomalous part of the video. Anomalous segments can then be detected by thresholding  $\rho(f)$ , or equivalently by thresholding  $P(y = +1|f)$  (Figure 2). This approach has the advantage of not relying on restrictive assumptions on a prior distribution of features because it works directly with the posterior distributions. For the same reason, it provides a natural reference decision threshold of 0.5, irrespective of the distribution of features across the videos. It also has the advantage of being a fast classifier and requiring constant time irrespective of the complexity of video. An effective and simple way of estimating  $\rho(f)$  is to estimate a logistic classifier from the data in  $R$  and  $I$ . Under the logistic model:

$$P(y|f; w) = \frac{1}{1 + e^{yw^T f}}, \quad (1)$$

where  $w$  is a vector of parameters estimated from the data. Specifically,  $w$  is obtained by maximizing the log-likelihood over the training data. Also, we add an L2 regularizer to help control over-fitting, resulting in the overall optimization problem:

$$\operatorname{argmax}_w \sum_i P(y_i|f_i; w) - \lambda \|w\|^2, \quad (2)$$

where the sum is taken over all the frames in the videos, and  $\lambda$  controls the regularization. We optimize it with stochastic gradient ascent with a decaying



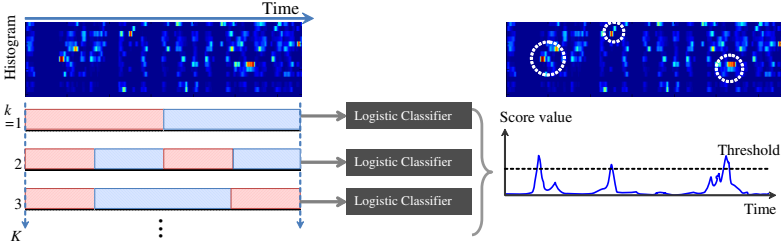
**Fig. 2.** Using density ratio estimation, e.g., logistic model, for detecting unusual events

learning-rate of  $\alpha_t = c/\sqrt{t}$ , where  $c$  is the step-size and  $t$  is the iteration number. In practice we use 100 iteration of the stochastic gradient outer loop. Importantly, this approach is entirely unsupervised.

## 2.2 Unsupervised Detection from a Single Input Video

The approach presented so far assumed two separate videos, one reference video showing nominal feature distribution and one input video in which we wish to detect the frames that are unusual with respect to the reference video. In our target task, however, we have a single input video from which we need to draw subsets of frames that can be used as reference/input pairs. More precisely, given an input video  $V$  with  $N$  frames, we separate it into two subsets  $V^+$  and  $V^-$  of equal sizes  $N/2$ .  $V^+$  and  $V^-$  are the analogs of  $R$  and  $I$  in the above introduction, except that they are drawn from a single video. We can then train a two-class classifier using a logistic model as described above, i.e., estimating  $w$  such that  $P(y|f; w)$  agrees with  $y = +1$  on  $V_+$  and  $-1$  on  $V_-$ . Those frames with feature vector  $f$  that occur frequently in both  $V_+$  and  $V_-$  will have a probability  $P(y|f; w)$  close to 0.5, while the unusual frames will have probability far from 0.5. This is of course the ideal case. In practice, however, the classifier is not perfect and an approach that is more robust to noise in the classifier is to use the median value  $M$  computed over the entire input  $V$  instead of 0.5 as reference value. We can then assign a score  $a_n$  to each frame  $n$  of  $V$  as:  $a_n = |P(y_n = +1|f_n; w) - M|$ .

Ideally, we should train a classifier and evaluate the scores on *all* the possible splits of  $V$ . Since this would require an impractically large number of rounds of logistic training, we limit ourselves to three splits corresponding to the following intervals of frames:  $V_+ = [1, N/2]$ ,  $V_+ = [N/4, 3N/4]$ ,  $V_+ = [1, N/4] \cup [N/2, 3N/4]$ . As shown in Figure 3, these three splits provide a good first order coverage of the possible splits of the data. From each split  $k$  we can estimate the parameter  $w^{(k)}$  of the logistic classifier of the corresponding binary classification problem, as described above, and for each frame  $n$  with feature  $f_n$  we can estimate the score  $a_n^{(k)} = |P(y_n^{(k)} = +1|f_n; w^{(k)}) - M^{(k)}|$ , where  $M^{(k)}$  is the median value of the probabilities over all the frames. The final score for each frame is obtained by averaging the scores:  $a_n = \sum_k a_n^{(k)}$  for frame  $n$ . The overall procedure for computing the scores is shown in Table 1. We implemented two ways of using the aggregate scores for detecting the interesting frames. The first approach,



**Fig. 3.** Top left: Visualization of the feature vector  $f$  computed at each frame. Middle left: Three different splits of the input video into +1/-1 classes. Bottom right: Final score obtained by averaging the outputs of the classifiers learned on the three splits (higher value = more interesting frames). Top right: The manually drawn white circles point to feature components that triggered detection of interesting events.

**Table 1.** Overall algorithm

Proposed Algorithm	
<b>Input</b> :	video sequence $V$ with $N$ frames with features $\{f_n\}$
<b>Output</b> :	$\{a_n, n = 1 \dots N\}$
<b>for</b>	$k=1, \dots, K$ <b>do</b>
1.	Generate a split $(V_+^{(k)}, V_-^{(k)})$
2.	Give label value to all $\{y_n^{(k)}\}$ based on
	$y_n^{(k)} = \begin{cases} +1 & (n \in V_+^{(k)}) \\ -1 & (n \in V_-^{(k)}) \end{cases}$
3.	Estimate parameter $w^{(k)}$ of logistic classifier from $\{f_n\}$ , and $\{y_n^{(k)}\}$
4.	Estimate the conditional probability $P(y_n^{(k)} = +1   f_n; w^{(k)})$ and median value $M^{(k)}$
5.	Calculate and accumulate the distance from median value
	$a_n = a_n +  P(y_n^{(k)} = +1   f_n; w^{(k)}) - M^{(k)} $
<b>end</b>	

labeled “Proposed1” in the result section, simply thresholds the scores so that frame  $n$  is retained if  $a_n > \epsilon$ . The second approach (“Proposed2”) is threshold free and is inspired from the classical SVM calibration procedure from Platt [17]. If  $M$  is the median value of  $a_n$  over all the frames in  $V$ , we define two subsets of frames,  $V_+^o$  and  $V_-^o$  corresponding to frames with scores below or above the threshold  $M$ , respectively. We can then estimate the parameter  $w_o$  of a logistic regressor for the split  $(V_+^o, V_-^o)$  and we obtain the final classification score by applying this logistic function to the original classification scores.

To compute the feature vector  $f$  of a frame, we first select  $p$  image points  $x_1, \dots, x_p$  in the frame and we compute a 576-dimensional descriptor  $\hat{f}(x_i)$  at each point. The set of  $\hat{f}(\cdot)$  computed over the entire video is quantized into  $K$  centers  $\hat{f}_j, j = 1, \dots, K$ . The final feature vector  $f$  used in the classifier is the  $K$ -dimensional histogram of quantized  $\hat{f}(\cdot)$  values computed over the video frame. Additional details are as follows:

**Point selection:** A standard approach to selecting feature points would be to use an interest point detector. We found that this technique generates too sparse a set of points for our approach. Instead, after coarse stabilization, we use all the points with intensity difference between consecutive frames greater than a threshold. Although simple, this approach yields a dense selection of points concentrated on the potentially interesting parts of the video.

$\hat{f}$ : We use a combination of histograms of gradient and flow vectors (HoG and HoF [18]), and motion boundary histograms (MBH [19]). We define a  $N \times N \times M$  patch around each  $x_i$ , which we divide into  $n_\sigma \times n_\sigma \times n_\tau$  cells. In each cell, we compute 1) a 8-bin histogram of gradient direction in that cell; 2) a 8-bin histogram of optical flow, using Farneback’s copencv implementation; 3) two 8-bin histograms encoding MBH (MBH uses histograms in the x and y axis, as in [19]). We use  $M=10$ ,  $N = 15$ ,  $n_\sigma = 3$ ,  $n_\tau = 2$  for a 576-dim descriptor at each point.

**Quantization:** We quantize  $\hat{f}$ : using  $K$ -means over the set of feature vectors from the entire video. We chose  $K = 32$  and verified that performance remains stable over a range of values 16–64 ( $K$  is kept constant across the experiments).

**Background stabilization:** Generally, personal videos tend to include shaky background motion because they are taken by hand-held cameras. This background motion affects the motion descriptors and the performance of the classifier. To minimize this effect, we estimate background motion by calculating a homography between consecutive frames and we align the frames prior to feature computation. We estimate the homography using LMeds by establishing KLT feature correspondences between frames. Since our event descriptors temporally span  $M = 10$  frames, we use stabilization over a 10 frame moving window.

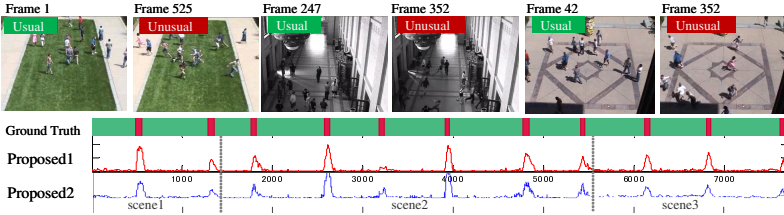
## 3 Experiments

### 3.1 Baseline Algorithms

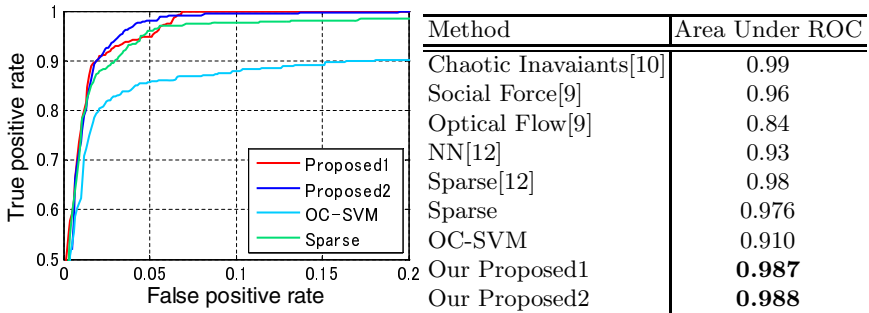
To test the effectiveness of our proposed algorithms, we use two baseline algorithms: One Class Support Vector Machine(OC-SVM), and sparse coding. These were chosen because of their good performance and because they are unsupervised techniques. OC-SVM is representative of outlier detection algorithms based on SVMs, which have produced excellent results in [20]. We used the publicly available implementation of [21], configured with a Gaussian kernel and  $\nu = 0.5$ . The second baseline is based on sparse reconstructability of query events from a learned dictionary, which is one of the state-of-the-art unusual event detection methods [12]. We used [22] for implementing sparse coding with Nesterov’s optimization method with a regularization parameter  $\lambda = 10$ .

### 3.2 UMN Dataset

We tested our proposed framework on the publicly available dataset of crowd videos from the UMN dataset [23]. This dataset consists of 11 different scenarios in 3 different scenes of crowd escape scenarios, over a total of 7740 frames. Each



**Fig. 4.** Example frames of usual (green) and unusual (red) events and the qualitative scoring results of our proposed methods for UMN dataset

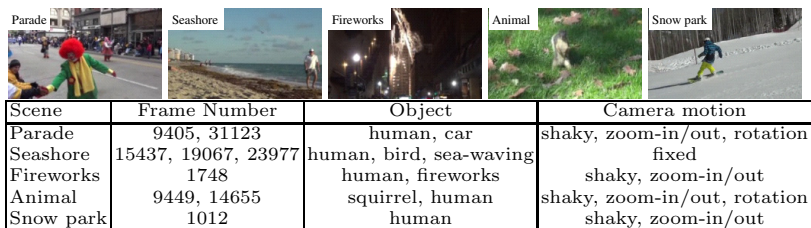


**Fig. 5.** ROC curve and performance comparison

video consists of an initial section of normal behavior and ends with sequences of unusual behavior (Figure 4). While these videos address a more specific task than our unconstrained detection problem, i.e., they are restricted to crowd motions and explicit “normal” section at the start of each video, they allow us to compare directly with published numbers in a way that does not favor our approach since many of the published techniques are tuned to crowd motions. In Figure 5, the AUC values of our methods outperform most of the other methods and are comparable to [10] and sparse coding [9]. However, our method is a more general solution, because it does not make any assumption about the content of the video, while [10] is specifically designed for anomalies in crowd videos, and [9] assumes that the first part of the video is nominal, i.e., can be used as reference to learn the dictionary, while we allow unusual events to occur anywhere in the video. Descriptor extraction takes about 0.43 second/frame for all the algorithms. Dictionary learning and classification takes 0.41, 0.022, and 0.020 second/frame for Sparse coding, OC-SVM, and our method, respectively, as measured on a single core 2.97 GHz Intel Core i7 PC with 8.0GB memory.

## 4 Personal Videos

We evaluated our framework using examples that are more representative of consumer videos. We used a dataset acquired in different scenes and locations using



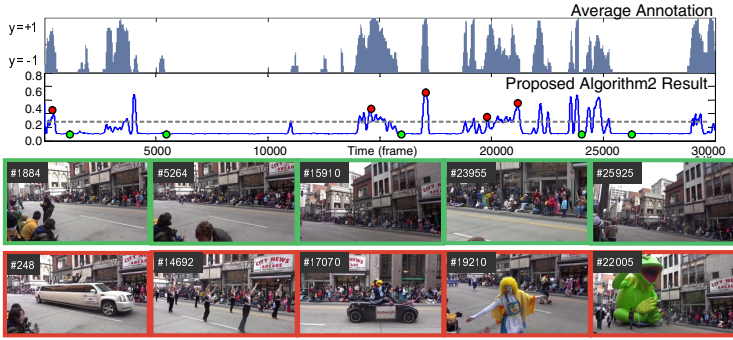
Scene	Frame Number	Object	Camera motion
Parade	9405, 31123	human, car	shaky, zoom-in/out, rotation
Seashore	15437, 19067, 23977	human, bird, sea-waving	fixed
Fireworks	1748	human, fireworks	shaky, zoom-in/out
Animal	9449, 14655	squirrel, human	shaky, zoom-in/out, rotation
Snow park	1012	human	shaky, zoom-in/out

**Fig. 6.** Personal Video Dataset. The dataset totally include 9 videos on 5 scene.

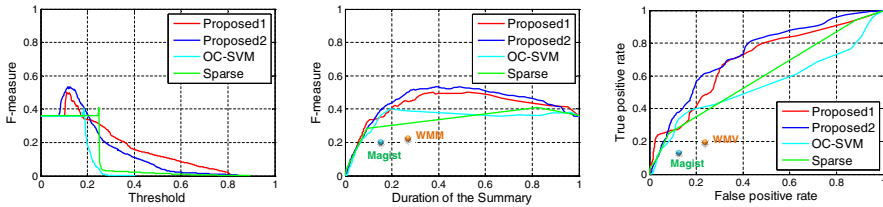
a hand-held consumer camera<sup>1</sup>. The videos include interesting events as well as long stretches of routine activity. The dataset consists of five different categories: parade, seashore, fireworks, animal, and snow park (Figure 6). In order to deal with variability in human annotations, we generated annotations of each video by fifteen different subjects. The annotators all received the same set of written instructions to detect rare and salient events in the entire video. To combine the multiple annotations, we compare the algorithms with each set of ground truth annotations and we average the resulting performance numbers across annotation sources. On average 17 % of the frames from the input videos are labeled as interesting. The average of all the 15 annotations is shown in Figure 7(top). Although the annotators disagree somewhat on the exact boundaries of the intervals of interest in the video, they do agree strongly on the general locations of the major events. A similar level of consensus is observed on all the annotations from all the videos. Quantitatively, the standard deviation of the length of video labeled as interesting relative to the length of the input video across all labelers is 5%. The score estimated by our Proposed2 algorithm is shown in Figure 7(middle) along with a few sampled frames detected as interesting or common by the algorithm are shown in Figure 7(bottom). In addition to SVM and sparse coding, we compared our proposed framework with two commercial products: Windows Movie Maker(WMM) and Magisto. Magisto [1] is one of most sophisticated video summarization services, which can automatically produce digested videos by using combinations of scene analysis and recognition algorithms. The scoring curve and the annotation averaged over the fifteen annotators are shown in Figure 7. The scores correlate well with human annotation data. It is interesting to note that, around ground truth events, the score decreases as the agreement among human annotators decreases. The overall performance is shown in the ROC and PR curves in Figure 8(b-c). For this dataset, chance performance is at precision 0.17 (maximum F-measure at 0.29.) In addition, Figure 8(a) compares classification performance as the detection threshold varies. This confirms that the performance of our proposed method gradually changes while maintaining higher F-measure value than the other algorithm. This implies that our method can be more easily tuned than the video summarization tools. Similar conclusions can be drawn from Table 2. Our approach outperforms other algorithms based on the area under the PR (average precision) or the ROC curves.

<sup>1</sup> The dataset is available at <https://sites.google.com/site/yitopaper/>





**Fig. 7.** Performance on one personal video: (Top) Average annotations from human labelers (+1 = interesting and -1 = common); (Middle) score returned by our algorithm (Proposed2) (higher score = more interesting frame); (Bottom) Sampled frames corresponding to the dots in the score curve (Green = common; red = interesting.)



**Fig. 8.** F-measure as function of detection threshold; (b) Precision/recall curve; (c) ROC curve

**Table 2.** Quantitative performance comparison

Algorithm	AU-FT	AU-FD	AU-ROC	Highest Precision(Duration)	Highest F-measure(Duration)
WMM	—	—	—	0.26 (0.23)	0.22 (0.23)
Magisto	—	—	—	0.44 (0.14)	0.21 (0.14)
OC-SVM	0.080	0.38	0.54	0.53 (0.09)	0.40 (0.19)
Sparse	0.098	0.36	0.60	0.43 (0.04)	0.41 (0.80)
Proposed1	<b>0.175</b>	0.42	0.69	0.53 (0.10)	0.51 (0.38)
Proposed2	0.152	<b>0.43</b>	<b>0.71</b>	<b>0.57</b> (0.15)	<b>0.54</b> (0.40)

For reference, we also indicate the highest F-measure and precision reached by each algorithm, along with the corresponding relative duration of the selected part of the video.

## 5 Conclusion

We proposed a feature-based event summarization method using an unsupervised logistic classifier framework for detecting frames which depart from the overall distribution of frames in the video. We showed promising performance on different types of datasets. In designing this approach, we deliberately limited ourselves to the distribution of low-level features in order to test the feasibility

of the method. However, these features may not be sufficient to discern subtle differences that make events unusual. One interesting direction is to combine high-level descriptors, e.g., including the responses of action detectors in the feature descriptor, with the current approach.

**Acknowledgement.** This work was partially funded by ARL under Agreement W911NF-10-2-0061.

## References

1. Magisto, <http://www.magisto.com>
2. Sugiyama, M., Yamada, M., von Bunaud, P., Suzuki, T., Kanamori, T., Kawanabe, M.: Direct density-ratio estimation with dimensionality reduction via least-squares hetero-distributional subspace search. *Neural Networks* 24 (2011)
3. Pritch, Y., Rav-Acha, A., Peleg, S.: Nonchronological video synopsis and indexing. *IEEE Trans. Pattern Anal. Mach. Intell.* 30, 1971–1984 (2008)
4. Kosmopoulos, D.I., Doulamis, N.D., Voulodimos, A.S.: Bayesian filter based behavior recognition in workflows allowing for user feedback. *Computer Vision and Image Understanding* 116, 422–434 (2012)
5. Li, Y., Lee, S.H., Yeh, C.H., Kuo, C.C.J.: Techniques for movie content analysis and skimming. *Signal Processing Magazine* 23, 79–89 (2006)
6. Nam, J., Tewfik, A.H.: Video abstract of video. In: *Proc. IEEE 3rd Workshop Multimedia Signal Processing*, pp. 117–122 (1999)
7. Jolic, N., Petrovic, N., Huang, T.: Scene generative models for adaptive video fast forward. In: *Proc. ICIP* (2003)
8. Xiong, Z., Radhakrishnan, R., Divakaran, A.: Generation of sports highlights using motion activity in combination with a common audio feature extraction framework. In: *Proc. ICIP*, vol. 1, pp. I-5–I-8 (2003)
9. Mehran, M.S.R., Oyama, A.: Abnormal crowd behavior detection using social force model. In: *Proc. CVPR* (2009)
10. Wu, S., Moore, B., Shah, M.: Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In: *Proc. CVPR* (2010)
11. Seo, H.J., Milanfar, P.: Static and space-time visual saliency detection by self-resemblance. *Journal of Vision* (2009)
12. Cong, Y., Yuan, J., Liu, J.: Sparse reconstruction cost for abnormal event detection. In: *Proc. CVPR* (2011)
13. Piciarelli, C., Micheloni, C., Foresti, G.: Trajectory-based anomalous event detection. *IEEE Transaction on Circuits and Systems for Video Technology* 18 (2008)
14. Zhang, D., Gatica-Perez, D., Bengio, S., McCowan, I.: Semi-supervised adapted hmms for unusual event detection. In: *Proc. CVPR* (2005)
15. Zhao, M., Saligrama, V.: Anomaly detection with score functions based on nearest neighbor graphs. In: *Proc. NIPS* (2009)
16. Matsugu, M., Yamanaka, M., Sugiyama, M.: Detection of activities and events without explicit categorization. In: *Proc. ICCV Workshop* (2011)
17. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: *Advances in Large Margin Classifiers* (1999)
18. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: *Proc. CVPR* (2008)

19. Dalal, N., Triggs, B., Schmid, C.: Human Detection Using Oriented Histograms of Flow and Appearance. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 428–441. Springer, Heidelberg (2006)
20. Scholkopf, B., Williamson, R., Smola, A., Taylor, J.S., Platt, J.C.: Support vector method for novelty detection. In: Proc. NIPS (2000)
21. Canu, S., Grandvalet, Y., Guigue, V., Rakotomamonjy, A.: Svm and kernel methods matlab toolbox. INSA de Rouen, Rouen, France (2005)
22. Matlab Toolbox,  
<http://www.mathworks.com/matlabcentral/fileexchange/16204>
23. UMN dataset, <http://mha.cs.umn.edu/Movies/>