On the Essence of Unsupervised Detection of Anomalous Motion in Surveillance Videos

Abdullah A. Abuolaim[1,2](✉), Wee Kheng Leow[1], Jagannadan Varadarajan[2], and Narendra Ahuja[2,3]

[1] Department of Computer Science,
National University of Singapore, Singapore, Singapore
{abdullah,leowwk}@comp.nus.edu.sg
[2] Advanced Digital Sciences Center, Singapore, Singapore
vjagan@adsc.com.sg
[3] Department of Electrical and Computer Engineering,
University of Illinois at Urbana-Champaign, Champaign, IL, USA
n-ahuja@illinois.edu

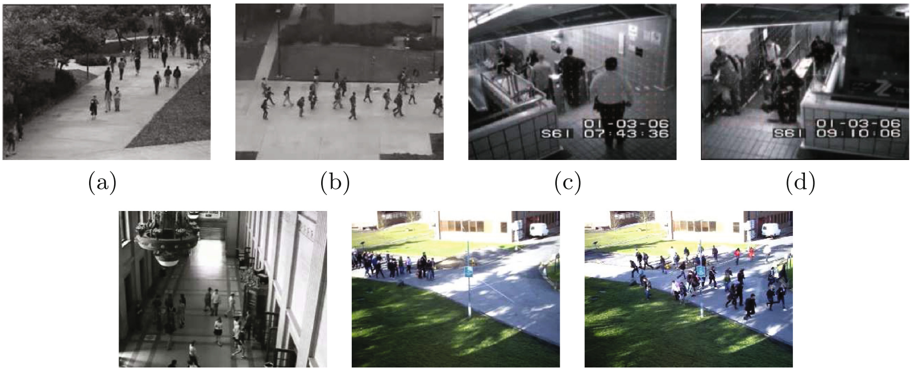
Abstract. An important application in surveillance is to apply computerized methods to automatically detect anomalous activities and then notify the security officers. Many methods have been proposed for anomaly detection with varying degree of accuracy. They can be characterized according to the approach adopted, which is supervised or unsupervised, and the features used. Unfortunately, existing literature has not elucidated the essential ingredients that make the methods work as they do, despite the fact that tests have been conducted to compare the performance of various methods. This paper attempts to fill this knowledge gap by studying the videos tested by existing methods and identifying key components required by an effective unsupervised anomaly detection algorithm. Our comprehensive test results show that an unsupervised algorithm that captures the key components can be relatively simple and yet perform equally well or better compared to existing methods.

1 Introduction

In recent decades, surveillance cameras have been widely used in public places to monitor human activities and provide security measures. A security officer typically has to monitor a dozen or more surveillance videos at the same time. Most of the time, there is no significant anomalous activity, which tends to lower the guard of the officer. After monitoring for long hours, he can get tired and miss important events that happen suddenly. Therefore, automatic detection of anomalous activities by computerized methods has attracted much research effort. These methods can also be used for criminal investigation to sieve through video archives to detect anomalous activities that have happened in the past.

Many methods have been proposed for anomaly detection with varying degree of accuracy. They can be characterized according to the approach adopted, which is supervised [1–16] or unsupervised [17–21], and the features used, which range from

M. Felsberg et al. (Eds.): CAIP 2017, Part I, LNCS 10424, pp. 160–171, 2017.
DOI: 10.1007/978-3-319-64689-3_13



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

(e) (f) (g)

Fig. 1. Sample frames from test videos. (a) UCSDped1, (b) UCSDped2, (c) Subway entrance, (d) Subway exit, (e) UMN, (f) PETS2009 scene 1, (g) PETS2009 scene 2.

low-level optical flow to high-level multiple object trajectories. Unfortunately, existing literature has not elucidated the essential ingredients that make the methods work as they do, despite the fact that tests have been conducted to compare the performance of various methods. For example, test results (Sect. 4) seem to suggest that there is no significant advantage in offline training performed by supervised methods compared to well-crafted unsupervised methods. It is also uncertain whether the time taken to process high-level features necessarily leads to better detection accuracy. This situation makes it difficult to optimize the methods for real-time online detection and efficient video archive analysis.

This paper attempts to fill this knowledge gap by studying the videos tested by existing methods and identifying key components required by an effective unsupervised anomaly detection algorithm. We have chosen to investigate unsupervised method instead of supervised method for the following reasons: (1) Unsupervised method does not require tedious and time-consuming manual labeling of training data. (2) It does not require an offline training phase. Therefore, it can be more easily extended to handle new normal and abnormal motion patterns that have not happened in the past. (3) Without the need of offline training, it can be more easily adapted to real-time online applications by implementing incremental algorithms. We focus on surveillance videos of pedestrians captured by stationary cameras because they are widely tested in the literature. Our comprehensive test results on these videos show that an unsupervised algorithm that captures the key components can be relatively simple and yet perform equally well or better compared to existing methods.

2 Existing Methods

Regardless of the approach, all existing methods begin by extracting features from the input videos and then making detection decisions based on the features.

The extracted features include optical flow [1,7,8,17,18], histogram of optical flow (HOF) [2,4,14–16,19,20], histogram of oriented gradient (HOG) [4,14], 3D SIFT [4,21], histogram of edge orientation [16], descriptors of intensity, gradient, object persistence, motion direction, optical flow orientation, speed, etc. [6,9,12], structural descriptors based on HOF [19], particle advection based on optical flow [3,13], tracked interest points or targets [3,7,19,20], dynamic texture [5,11], and pedestrian regions [19]. In addition, auto-encoder neural network has also been used to extract features from video images [1,10,14]. These features may be extracted for image pixels [3,7,13,17], 2D spatial regions [1,2,5,6,8,10–12,15,16,18–20] or 3D spatio-temporal regions [4,6,9,14,15,21] of the video. Simple features, such as optical flow and intensity gradient, take much less time to extract compared to features extracted by complex algorithms, such as pedestrian detection and multiple target tracking [19,20]. Auto-encoders can extract features efficiently but it takes a large amount of time to train them.

Existing methods for detecting anomalous motion in surveillance videos can be grouped into two categories: supervised and unsupervised. Supervised methods [1–16] typically work in two phases: training and testing. In the training phase, these methods use labeled training data to train a classifier or a probabilistic model. Various algorithms have been used for training, including SVM [1,7], conjugate Bayesian analysis [2], EM [3,5,11–13], Gaussian process regression [4], Bayesian network propagation [6], recurrent neural network [10], and sparse reconstruction [15]. Methods that use k-nn [9,16] do not need the training phase. Methods that model simple probability distributions such as Gaussian distributions [8,14] have a simple training phase that estimates the distribution parameters. In the testing phase, trained classifier or probabilistic model is used to classify features as normal or abnormal. Well-trained supervised methods can be accurate. Moreover, their testing phases are typically efficient enough for real-time applications, provided the features can be extracted efficiently. However, manual labeling of training data is tedious and time-consuming. Therefore, it is difficult to extend supervised methods to include new scenario.

Unsupervised methods [17–21] typically group extracted features into clusters without relying on labeled data. The clustering algorithms that have been used include hierarchical cluster merging [18], k-means [20], online weighted clustering [20], and fuzzy probabilistic clustering [21]. After clustering, these methods label dominant clusters (i.e., clusters with the most members) as normal and the other clusters as abnormal. The threshold for deciding which clusters are dominant is empirically set. The methods of [17,19], on the other hand, do not perform clustering. Instead, the method of [17] performs line intersection to detect the center of crowd dispersion, and the method of [19] measures dissimilarity between features to detect anomalies. Unsupervised methods do not require manually labeled training data and do not perform offline training. Therefore, they can be easily extended to handle new normal and abnormal motion. Moreover, unsupervised methods that use incremental algorithms are very suitable for real-time online applications.

The above methods have been tested on one or more of the following surveillance videos on pedestrians (Fig. 1):

- UCSDped1 [22]: 36 videos, tested in [1,4–6,8–13,15,16,19–21].
- UCSDped2 [22]: 12 videos, tested in [5,8,11–16,19–21].
- Subway [8]: 2 videos in 2 scenes, tested in [4–6,8,9,12,15,21].
- UMN [23]: 3 videos in 3 scenes, tested in [2,3,5,7,9,13–15,17–19].
- PETS2009 [24]: 8 videos in 2 scenes, tested in [2,3,13,18].

For UCSDped1, UCSDped2, UMN, and PETS2009, the walking pedestrians constitute the dominant motion and they are regarded as normal. Abnormal motion is elicited by carts, cyclists, skaters, escaping humans, etc., which move at faster speeds. That is, normal and abnormal motion in these videos differ primarily in motion speed. On the other hand, for Subway, the passengers entering and existing the subway gates in an orderly manner constitute the dominant motion and are regarded as normal. Passengers who move along directions other than entering or exiting the gates are regarded as abnormal. That is, normal and abnormal motion in these videos differ primarily in motion direction.

3 Unsupervised Anomaly Detection

Our research goal is to identify the essential ingredients for effective unsupervised detection of anomalies in pedestrian surveillance videos. To achieve this goal, we apply the principle of Occam's razor: given several equally effective alternatives, we choose the simplest alternative. Therefore, we call our method OCCAM. Similar to unsupervised methods based on clustering, OCCAM consists of three stages: (1) feature extraction, (2) features clustering, and (3) cluster labeling.

3.1 Feature Extraction

Analysis of common test videos used in existing work (Sect. 2) shows that normal and abnormal motion may be differentiated by either motion speed or motion direction alone, depending on the test videos. Therefore, OCCAM uses motion speed or motion direction as the feature. It applies the method of [25] to extract trajectories of distinctive image feature points. This method samples feature points at multiple spatial scales and tracks feature points using median filtering to obtain optical flow. Stationary feature points and those with large displacements between two consecutive frames are removed to reduce tracking error. Tracked feature trajectories have a fixed length l, and long trajectories are split into short trajectories of length l. Trajectories with length shorter than l are removed because they are insignificant.

Let $\{\mathbf{x}_i(t), \ldots, \mathbf{x}_i(t{+}l)\}$ denote the trajectory of feature point p_i, $i = 1, \ldots, n$, from frame t to $t + l$, where $\mathbf{x}_i(t)$ is the position of p_i in frame t. Then, the direction θ_i and speed s_i of feature point p_i are computed as the direction and magnitude divided by trajectory length of the vector $\mathbf{x}_i(t + l) - \mathbf{x}_i(t)$.

In UCSDped1 videos, humans and other objects move toward or away from the camera resulting in noticeable perspective distortion. As a result, objects nearer to the camera appears to move faster than those further from the camera even though they may move at the same actual speed. To overcome this distortion, the feature points are projected onto the ground plane using an estimated homography. Then, the speeds of the feature points are computed after projection.

3.2 Feature Clustering

Feature clustering is performed on either motion speed or motion direction. Let us denote the extracted feature values as f_i, $i = 1, \ldots, n$. Since the features are 1-D, the simplest way to cluster f_i is to divide the feature value range (minimum to maximum) into m equal intervals, and regard each interval as a cluster C_j, $j = 1, \ldots, m$. Then, features f_i can be clustered efficiently into their respective clusters in a fixed $O(n)$ time. Each cluster C_j is characterized by the cluster size $|C_j|$ and the cluster center, which is the average feature value $\bar{f}_j$ of the features in C_j. This simple and efficient clustering method ensures that the intra-cluster differences are much smaller than the inter-cluster differences.

After clustering, normalized cluster size S_j and normalized cluster center F_j are computed for each cluster C_j. Let us denote the dominant cluster, the cluster with the largest size, as C^+ and the largest feature value as f^*. Then, S_j and F_j are computed as follows:

$$S_j = |C_j|/|C^+|, \quad F_j = \bar{f}_j/f^*. \tag{1}$$

Therefore, these normalized values range between 0 and 1. Each cluster C_j is now characterized by a characteristic vector of two components, namely normalized cluster size S_j and normalized cluster center F_j.

3.3 Cluster Labeling

Unlike existing methods, OCCAM labels the clusters into three types: normal, abnormal, and ambiguous. The ambiguous clusters allow the normal and abnormal clusters to be separated as widely as possible. Since the characteristic vectors of the clusters are 2-D, 2-D k-means clustering is used to group the clusters C_j into three groups G_h, $h = 1, 2, 3$.

First, k-means clustering is initialized as follows: The center of group G_1 is initialized as the characteristic vector of the dominant cluster C^+. Similarly, the abnormal group G_2 is initialized with the cluster C^- whose cluster center is the furthest from that of C^+ because C^- is most likely to be abnormal. The ambiguous group G_3 is initialized with the cluster that is approximately equidistant to C^+ and C^-.

Next, k-means clustering is executed to group the remaining clusters C_j into the three groups G_h. The distance between a cluster and a group is measured in terms of the Euclidean distance between their characteristic vectors. After clustering, all the clusters in group G_1 are labeled as normal, those in G_2 abnormal, and those in G_3 ambiguous. In addition, the abnormal cluster that is nearest to G_1 is re-labeled as ambiguous so as to widen the separation between normal and abnormal clusters.

After cluster labeling, the features f_i in abnormal clusters are labeled as abnormal features. The corresponding trajectory positions $\mathbf{x}_i(t)$ of f_i are labeled as abnormal feature points. Finally, the video frames that contain abnormal feature points are labeled as abnormal frames.

4 Experiments and Discussions

4.1 Data Preparation and Procedure

Five sets of common test videos discussed in Sect. 2 were used in the experiments, namely UCSDped1, UCSDped2, Subway, UMN, and PETS2009. For OCCAM, motion directions were extracted from Subway video whereas motion speeds were extracted from the other videos. Next, feature clustering and cluster labeling were performed to detect abnormal feature points and abnormal frames. Then, true positive rate (TPR), false positive rate (FPR), and accuracy of detected abnormal frames were computed.

To determine a suitable value for the number of clusters m in the feature clustering stage, a test was performed on one video each from UCSDped1, UCSDped2, PETS2009 scene 1 and PETS2009 scene 2 test sets with varying values of m. The test shows that OCCAM achieves the overall highest accuracy with $m = 10$. Therefore, m is set to 10 for all the tests.

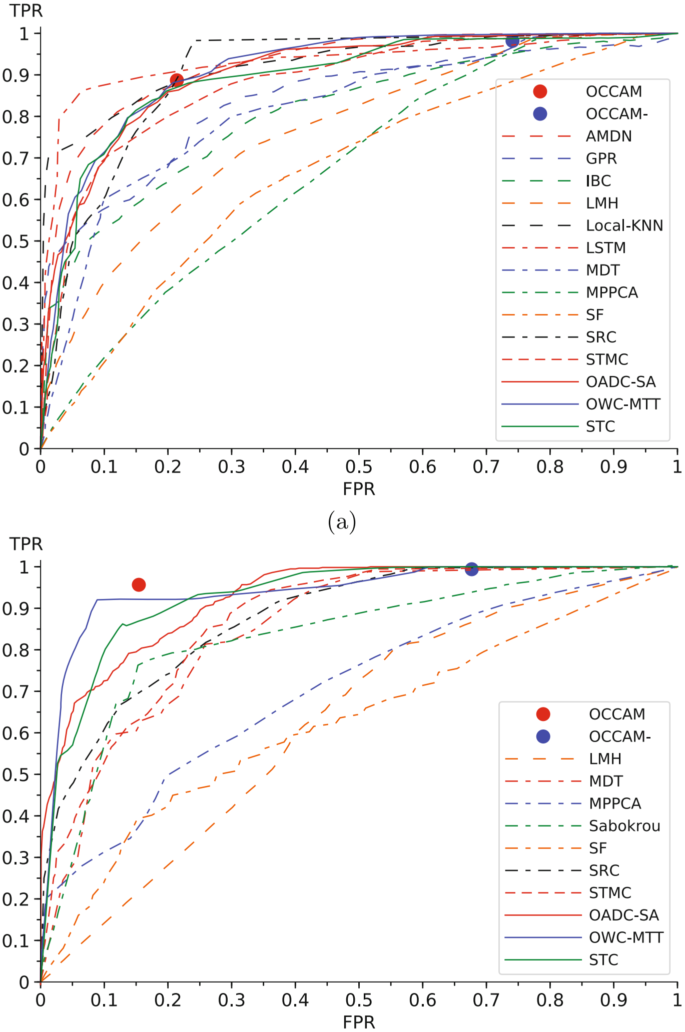
4.2 Benefit of Ambiguous Clusters

This test illustrates the benefit of having ambiguous clusters. A variant of OCCAM, denoted as OCCAM−, was tested such that its cluster labeling stage ran k-means clustering with $k = 2$ for normal and abnormal groups, without ambiguous group. Existing methods also label their clusters as either normal or abnormal, without ambiguous clusters. Both OCCAM and OCCAM− were tested on the common test videos discussed in Sect. 2. True positive rate (TPR) and false positive rate (FPR) were measured for the detected abnormal frames.

Table 1 compares the results of OCCAM and OCCAM−. For all test videos, OCCAM's TPR is slightly smaller than that of OCCAM−, but OCCAM's FPR is significantly smaller than that of OCCAM−. That is, by regarding some clusters as ambiguous, OCCAM makes significantly fewer false detections than does OCCAM− without significantly sacrificing its true detection rate.

4.3 Performance Comparison

OCCAM's results are compared with all of the existing methods discussed in Sect. 2. These methods belong to the following categories:

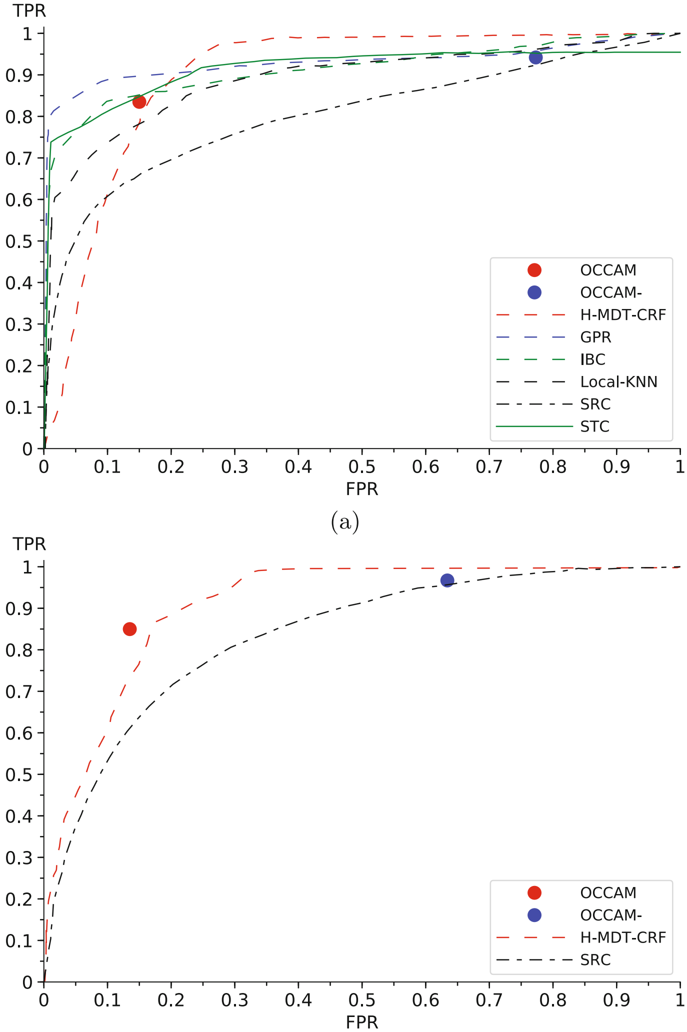


(a)

(b)

Fig. 2. Performance comparison. 14 methods are available for comparison on (a) UCSDped1 videos and 10 methods on (b) UCSDped2 videos. Supervised methods (dashed lines), unsupervised methods (solid lines).

- Supervised: AMDN [1], BM [2], CI [3], GPR [4], H-MDT-CRF [5], IBC [6], IEP [7], LMH [8], Local-KNN [9], LSTM [10], MDT [11], MPPCA [12], OF [13], SF [13], Sabokrou [14], SRC [15], and STMC [16]. [13] tested both OF and SF methods.
- Unsupervised: DC [17], FF [18], OADC-SA [19], OWC-MTT [20], and STC [21].



(a)

(b)

Fig. 3. Performance comparison. 6 methods are available for comparison on (a) Subway entrance video and 2 methods on (b) Subway exit video. Supervised methods (dashed lines), unsupervised methods (solid lines).

Most of these methods were tested only on some of the test videos. The test results on UCSDped1, UCSDped2, and Subway were reported as ROC curves. For the test results on UMN, some papers reported ROC curves whereas others reported only accuracy. For PETS2009, only accuracy was reported. ROC curves are not reported for H-MDT-CRF [5] on UCSDped1 and UCSDped2, LMH [8] and MPPCA [12] on Subway, and Sabokrou [14] on UMN. Therefore, they are

Table 1. Benefit of ambiguous clusters. OCCAM (O) has slightly smaller TPR, but significantly smaller FPR compared to OCCAM− (O−).

Test videos	TPR		FPR	
	O	O−	O	O−
UCSDped1	0.887	0.982	0.214	0.741
UCSDped2	0.957	0.994	0.154	0.677
Subway Entrance	0.835	0.942	0.152	0.773
Subway Exit	0.850	0.967	0.136	0.634
UMN	0.910	0.999	0.002	0.818
PETS2009 Scene 1	0.892	0.973	0.079	0.482
PETS2009 Scene 2	0.987	0.999	0.125	0.395

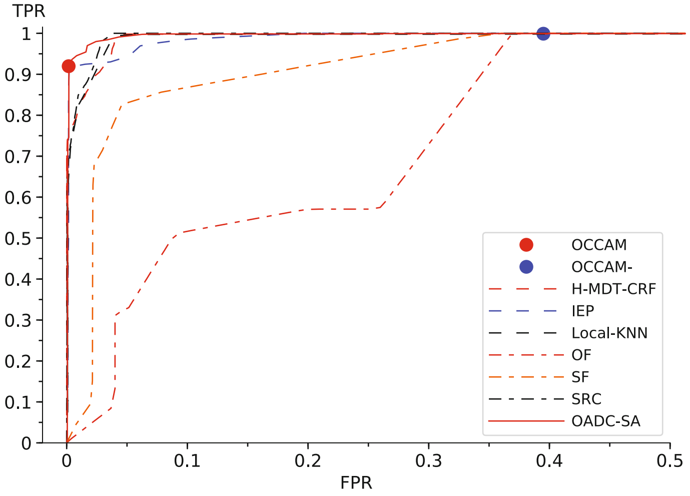


Fig. 4. Performance comparison on UMN video. 7 methods are available for comparison. Supervised methods (dashed lines), unsupervised methods (solid lines).

not included in our ROC graphs. The ROC curves reported in this paper are plotted using either the test results provided by the authors or a software that traces the curves' points presented in existing papers.

For UCSD (Fig. 2) and UMN videos (Fig. 4), OCCAM is among the best performers compared to existing methods. For the Subway videos (Fig. 3), OCCAM's performance is comparable to those of existing methods that are far more complex than OCCAM. For the same FPR, OCCAM achieves the highest TPR compared to existing methods for UCSDped2 (Fig. 2b), Subway exit (Fig. 3b), and UMN (Fig. 4), the 3rd highest TPR for UCSDped1 (Fig. 2a), and the 4th highest TPR for Subway entrance (Fig. 3a). In applications where high FPR is tolerable, OCCAM can run as OCCAM− without ambiguous clusters.

Table 2. Performance comparison on UMN and PETS2009 videos. OCCAM has the highest overall accuracy. (S) Supervised method, (U) unsupervised method.

Method	Type	UMN	PETS2009	PETS2009
			scene 1	scene 2
OCCAM	U	**0.98**	**0.91**	**0.99**
BM [2]	S	0.96	0.89	0.94
CI [3]	S	0.88	0.60	0.93
SF [13]	S	0.85	0.59	0.85
SRC [15]	S	0.85	–	–
DC [17]	U	0.96	–	–
FF [18]	U	0.81	0.38	0.88

Then, OCCAM− achieves TPR of close to 1.0 for all test cases. Figures 2, 3 and 4 also show that existing unsupervised methods can perform as well as or better than supervised methods.

Some existing papers reported only accuracy on UMN and PETS2009 videos. Table 2 shows that OCCAM is more accurate than these methods for both UMN and PETS2009.

For UCSDped1 and UCSDped2 videos, Li and Mahadevan [5,11] also proposed a pixel-level criterion to measure the spatial accuracy of detected abnormal frames. This error measure depends on the number of detected abnormal pixels in an abnormal region. Since OCCAM detects only selected pixels in these regions instead of the whole regions, pixel-level criterion is not appropriate for OCCAM. Instead, this paper measures spatial accuracy in terms of precision, which is the percentage of detected abnormal pixels that are true positives. OCCAM achieves abnormal pixel detection precision of 0.72 for UCSDped1 and 0.78 for UCSDped2. Moreover, most of the false positive pixels are located around the abnormal regions. On the other hand, the spatial precision of OCCAM− on UCSDped1 and UCSDped2 is, respectively, 0.37 and 0.40, which is much lower than that of OCCAM. Therefore, ambiguous clusters are important for OCCAM to achieve high spatial accuracy in detecting abnormal pixels.

5 Conclusions

This paper investigated the essential components required for effective unsupervised detection of anomalies in surveillance videos of pedestrians. It shows that relatively simple but well-designed unsupervised algorithm like OCCAM can perform as well as or better than existing supervised and unsupervised methods. In particular, simple but informative features such as motion direction and motion speed are sufficient for achieving high TPR with low FPR. Moreover, inclusion of ambiguous clusters in the cluster labeling process reduces FPR significantly without sacrificing TPR much. At the same FPR, OCCAM achieves among the highest TPR compared to existing methods. It also has the highest accuracy for UMN and PETS2009 videos compared to existing methods that reported only accuracy. In applications where high FPR is tolerable, OCCAM can run as OCCAM− without ambiguous clusters. Then, OCCAM− achieves TPR of close to 1.0 for all test cases. With ambiguous clusters, OCCAM's spatial precision of detecting abnormal pixels is also very high. In general, OCCAM and existing unsupervised methods can perform as well as or better than supervised methods. Therefore, our research results can serve as a useful benchmark for testing new algorithms and for developing more advanced algorithms that require features other than motion speed and direction.

References

1. Xu, D., Ricci, E., Yan, Y., Song, J., Sebe, N.: Learning deep representations of appearance and motion for anomalous event detection. In: Proceedings of the BMVC, pp. 1–12 (2015)
2. Wu, S., Wong, H.S., Yu, Z.: A Bayesian model for crowd escape behavior detection. IEEE Trans. Circ. Syst. Video Technol. **24**(1), 85–98 (2014)
3. Wu, S., Moore, B.E., Shah, M.: Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In: Proceedings of the CVPR, pp. 2054–2060 (2010)
4. Cheng, K., Chen, Y., Fang, W.: Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression. In: Proceedings of the CVPR, pp. 2909–2917 (2015)
5. Li, W., Mahadevan, V., Vasconcelos, N.: Anomaly detection and localization in crowded scenes. IEEE Trans. PAMI **36**(1), 18–32 (2014)
6. Boiman, O., Irani, M.: Detecting irregularities in images and in video. IJCV **74**(1), 17–31 (2007)
7. Cui, X., Liu, Q., Gao, M., Metaxas, D.N.: Abnormal detection using interaction energy potentials. In: Proceedings of the CVPR, pp. 3161–3167 (2011)
8. Adam, A., Rivlin, E., Shimshoni, I., Reinitz, D.: Robust real-time unusual event detection using multiple fixed-location monitors. IEEE Trans. PAMI **30**(3), 555–560 (2008)
9. Saligrama, V., Chen, Z.: Video anomaly detection based on local statistical aggregates. In: Proceedings of the CVPR, pp. 2112–2119 (2012)
10. Feng, Y., Yuan, Y., Lu, X.: Deep representation for abnormal event detection in crowded scenes. In: Proceedings of the ACM MM, pp. 591–595 (2016)
11. Mahadevan, V., Li, W., Bhalodia, V., Vasconcelos, N.: Anomaly detection in crowded scenes. In: Proceedings of the CVPR, vol. 249, p. 250 (2010)
12. Kim, J., Grauman, K.: Observe locally, infer globally: a space-time MRF for detecting abnormal activities with incremental updates. In: Proceedings of the CVPR, pp. 2921–2928 (2009)
13. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using social force model. In: Proceedings of the CVPR, pp. 935–942 (2009)
14. Sabokrou, M., Fathy, M., Hoseini, M., Klette, R.: Real-time anomaly detection and localization in crowded scenes. In: Proceedings of the CVPR Workshops, pp. 56–62 (2015)
15. Cong, Y., Yuan, J., Liu, J.: Sparse reconstruction cost for abnormal event detection. In: Proceedings of the CVPR, pp. 3449–3456 (2011)

16. Cong, Y., Yuan, J., Tang, Y.: Video anomaly search in crowded scenes via spatio-temporal motion context. IEEE Trans. Inform. Forensics Secur. **8**(10), 1590–1599 (2013)
17. Chen, C.Y., Shao, Y.: Crowd escape behavior detection and localization based on divergent centers. IEEE Sens. J. **15**(4), 2431–2439 (2015)
18. Chen, D.Y., Huang, P.C.: Motion-based unusual event detection in human crowds. J. Vis. Commun. Image Representation **22**(2), 178–186 (2011)
19. Yuan, Y., Fang, J., Wang, Q.: Online anomaly detection in crowd scenes via structure analysis. IEEE Trans. Cybern. **45**(3), 548–561 (2015)
20. Lin, H., Deng, J.D., Woodford, B.J., Shahi, A.: Online weighted clustering for real-time abnormal event detection in video surveillance. In: Proceedings of the ACM MM, pp. 536–540 (2016)
21. Roshtkhari, M.J., Levine, M.D.: Online dominant and anomalous behavior detection in videos. In: Proceedings of the CVPR, pp. 2611–2618 (2013)
22. UCSD: Anomaly Detection Dataset. www.svcl.ucsd.edu/projects/anomaly/dataset.htm
23. UMN: Unusual Crowd Activity Dataset. www.mha.cs.umn.edu/proj_events.shtml
24. PETS2009: Event Recognition Dataset. www.cvg.reading.ac.uk/PETS2009/a.html#s3
25. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: Proceedings of the CVPR, pp. 3169–3176 (2011)