# FAST ANOMALY DETECTION IN TRAFFIC SURVEILLANCE VIDEO BASED ON ROBUST SPARSE OPTICAL FLOW

*Hanlin Tan, Yongping Zhai, Yu Liu and Maojun Zhang*

College of Information System and Management, National University of Defense Technology, China
hanlin_tan@nudt.edu.cn, zhaiyp@visionsplend.com, jasonyuliu@nudt.edu.cn, zmjbar@163.com

## ABSTRACT

Fast abnormal events detection in video is important for intelligent analysis of video. This paper proposes a fast anomaly detection algorithm based on sparse optical flow. We improve the efficiency of optical flow computation with foreground mask and spacial sampling and increase the robustness of optical flow with good feature (TK) points selecting and forward-backward filtering. A foreground channel is also added to the feature vector to help detect static or low speed objects. The algorithm is validated on real-life traffic surveillance to prove its effectiveness. It is also evaluated on a benchmark dataset and achieve detection results comparable to state-of-art methods and outperforms them at pixel-level when the false alarm rate is low. The strength of our algorithm is that it runs real-time on the benchmark dataset which is hundreds of times faster than comparative methods.

***Index Terms***— Anomaly detection, abnormal event, traffic surveillance, optical flow

## 1. INTRODUCTION

Automatically locating abnormal events in traffic surveillance video is of vital importance to traffic administration as well as public safety. As is well known, it is hard to handle all scenes with one method, so we select traffic surveillance as the target type of scene to design and test our algorithm.

### 1.1. Definition of Anomaly

There are many definitions of anomaly to many people. We take events with low possibility as anomaly since it converts the ambiguous concept to an operational one. The word *event* is still not operational. This word contains different meanings in different scenes. In traffic surveillance, we take event as motion. Therefore, anomaly detection in this paper is to detect motion with low possibility in traffic surveillance video. This definition ignores anomaly that is not involved with motion, e.g. appearance anomaly. However, this is acceptable considering our application background being traffic surveillance. The definition is consistent with human cognition in our application and it introduces what features to extract in Section 2.

### 1.2. Related Work

A group of anomaly detection algorithms[2, 3, 4, 1, 5, 6, 7, 8] performs (subsets of) the following five steps:

a) Feature computation on pixel level;
b) Feature aggregation in space or/and time;
c) Transformation of the aggregated features to certain domains;
d) Build a model/classifier with the final features from training video;
e) Comparison of the final features from test video with the model.

Pixel level features include foreground location[4, 7, 8], HOG (Histogram of Oriented Gradients)[1], HOF (Histogram of Optical Flow)[2] and MDT (Mixtures of Dynamic Textures)[3]. The most common aggregation is to sum up features in a spacial and temporal 3-D block, which helps make the feature more robust to noise. Other aggregation includes building custom models such as locality model used in [8]. Transformation used recently includes sparse representation[1, 9]. Models and classifiers include sparse reconstruction cost[1, 6], maximum norm[7] and one-class SVM[5]. For detection step, researchers usually have to set thresholds or tune parameters based on what model or classifier they adopts and compare features they extract from training and test videos.

Another type of method is based on tracking[10]. This type of method is good at handling uncrowded scenes. However, tracking is unreliable on crowded scenes[3]. And it is hard to obtain reliable detection results with unreliable trajectories. The algorithm proposed by this paper belongs to the first type and can achieve remarkable detection results on real-life traffic surveillance in real-time.

### 1.3. Our Contributions

The contribution of this paper to anomaly detection lies in several aspects: First, it proposes a procedure comput-

ing robust sparse optical flow, which makes feature extraction fast and reliable. Second, it aims at real-life traffic surveillance and propose a framework that is simple, robust and hundreds of times faster than comparative methods. Third, it summarizes a common framework the state-of-art methods adopt. The source code of our algorithm is available at https://github.com/TomHeaven/AnomalyAnalysisWithOpticalFlow.

## 2. METHODOLOGY

This section first introduces how to compute robust sparse optical flow, and then elaborate how to extract features and utilize it to detect anomaly.

### 2.1. Robust Sparse Optical Flow

Figure 1 (a) illustrates the result of the optical flow, which is in accord with object motion. The computation of optical flow is mainly based on [11, 12, 13]. They are integrated with modifications and achieve improvements on both computation speed and robustness. The procedure is illustrated as Figure 1 (b). First, the foreground-masked input frame owns fewer and only moving pixels, which reduce the amount of calculation as well as the chance of matching errors. Second, finding good features makes our optical flow more reliable[12]. Third, the LKT tracker[11] is the most commonly used and stable method for computation of optical flow. LKT tracker is used to compute optical flow only on good feature points[12], which is both robust and fast. At last, a forward-backward filter inspired by [13] further removes the unreliable matching results and leaves us the robust optical flow. The optical flow is computed in both directions and the distance between the origin of forward flow and the destination of the backward is recorded. Then the worst 50% optical flow is filtered out by a mean filter. Note the original thesis[13] use a median filter. However, we find that a mean filter performs better in traffic videos.

### 2.2. Feature Extraction and Aggregation

Figure 1 (c) illustrates the procedure for feature extraction and aggregation. One of the low-level features is optical flow. It captures both the speed and the direction of every moving pixel. Then the optical flow is projected on a certain number of orientations to obtain the Histogram of Optical Flow (HOF) feature. The HOF feature is aggregated in spatial block and temporal period by sum them up in each orientation separately. ([1] called this feature Multi-scale HOF, MHOF.)

However, one important limitation of optical flow is that it cannot extract feature from static or slow speed object, even if the object is detected as foreground by motion detection algorithms. In fact, computation of optical flow requires the corresponding pixels' distance between the two input frames to be in a certain range. If the pixel distance is too small, the length of optical flow is close to zero and cannot be computed. If the pixel distance is too large, the algorithm cannot find the correct corresponding pixel to compute optical flow. In order to make up this limitation, a new channel named *foreground* is added to feature inspired by [4]. For each pixel,

$$foreground = \begin{cases} 1, & \text{for foreground pixel} \\ 0, & \text{for background pixel} \end{cases} \quad (1)$$

The aggregation step for this new channel is identical to that for the HOF feature. With this new channel, the feature has better performance on detecting static and low speed foreground object than simply using HOF.

To further improve robustness, a spacial Gaussian blur is performed on the aggregated feature (on each channel separately), which makes the feature more smooth and stable. The experimental results show that the blur of feature does not only reduce false alarm but also increase true positive detection rate.

### 2.3. Training and Detection

A training model is built to learn from a normal video and detect anomaly in a test video. Figure 1 (d) and (e) illustrates one block of the proposed feature for some 5,000 normal frames extracted from a piece of real-life traffic surveillance video. The values of feature channels vary dramatically, yet they are extracted from the normal video. A very simple detection criterion is that:

a) Any feature value that appear in training video are normal.
b) Any feature value that is significantly greater than the maximum feature values of training video are considered abnormal.

Let vector $A(b,t)$ denote the aggregated feature of block $b$ at frame $t$. The training process is to find a maximum boundary $B(b)$ for each feature channel:

$$B(b) = \max_t A(b,t) \quad (2)$$

where $t$ is a variable that enumerates all frames of training sequences. Let vector $v(b,t)$ denote the aggregated feature extracted from the test video. Then, compute the distance vector $D(b,t)$ as

$$D(b,t) = v(b,t) - B(b) \quad (3)$$

Finally, whether a block is abnormal is decided by thresholding on each channel of the distance vector:

$$x(b,t) = \begin{cases} 1, & if D(b,t) > \theta \\ 0, & otherwise \end{cases} \quad (4)$$

where $\theta$ is the predefined threshold vector.

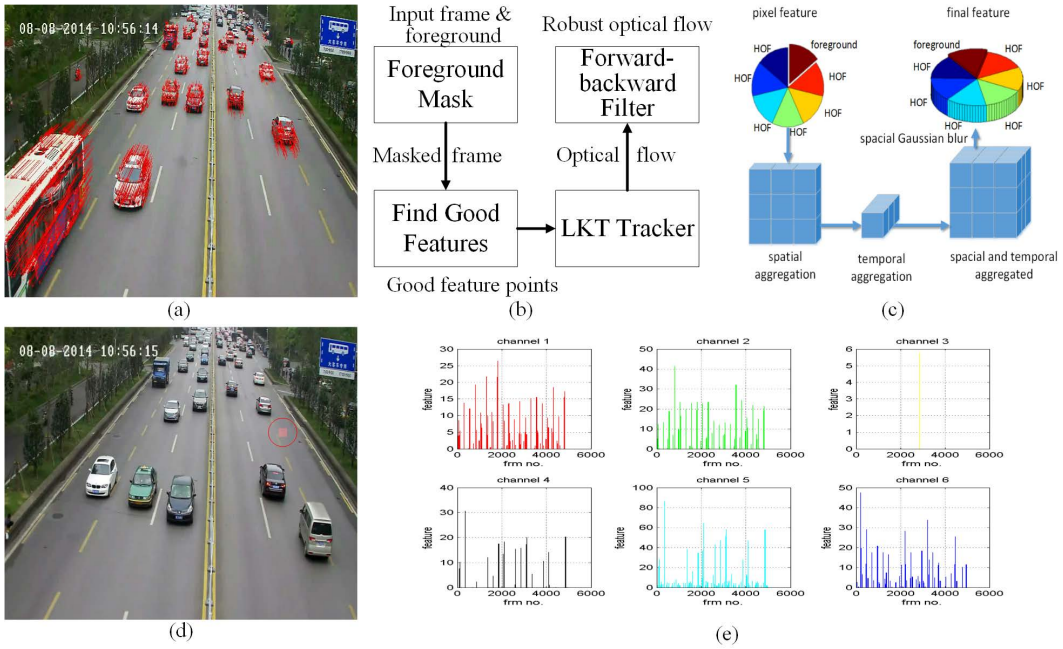



**Fig. 1**. Optical flow computation and feature extraction. (a) illustrates the Extracted sparse optical flow. The head and length of red arrows denotes the direction and speed of extracted optical flow. (b) demonstrates four major steps to compute robust optical flow with high speed. (c) illustrates feature extraction and aggregation. Pixel feature consists of HOF with additional channel of foreground [4]. To further improve robustness, a spacial Gaussian blur is performed on the aggregated feature (on each channel separately). (d) demonstrates a scene image and the red block circled marks out one of the feature blocks. (e) illustrates its feature variation along time with a plot of six channels of the feature. The horizontal axis is the frame number; the vertical axis is the feature value.

### 2.4. Acceleration

The most time-consuming parts of our algorithms lie in the computation of optical flow and aggregation of features. For optical flow, there are two methods for acceleration:

a) Foreground mask. Computation of optical flow in the background area is neither nonsense nor wrong. Therefore, foreground is used as a mask to increase both efficiency and robustness.
b) Spacial sampling. Sparse optical flow extracted from fixed grid pixels are sufficient for feature aggregation.

For aggregation, integral images are used to compute the sum of a rectangle area with a constant time cost.

## 3. EXPERIMENTS

The method is evaluated on both real-life situations and a benchmark dataset. Parameter setting are as follows: the block size is $16 \cdot 16$ with a time window of 5 frames. The threshold vector is different according to scenes. However, $0.1 \cdot num\_of\_block\_pixels / spacial\_sample\_distance \cdot time\_window$ typically produces an acceptable result.

### 3.1. Real-life Situations

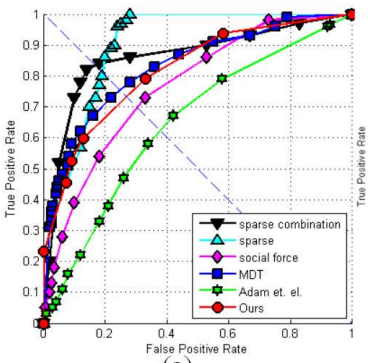
The proposed method is tested in numerous real-life situations to validate its effectiveness. Figure 2 illustrates the results. Given a few minutes of normal video for training, our algorithm accurately detect various anomalies such as pedestrian across the road at wrong location and reversely running motorcycles and trunks at different scenes, day and night.

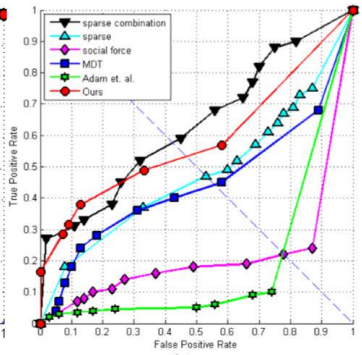
### 3.2. UCSD Ped1 Dataset

The algorithm is tested on one of the most evaluated datasets: UCSD Ped1 Dataset [15]. This dataset provide training sequences with only pedestrians and marks non-pedestrians as anomalies. Note the ground truth is not fully consistent with our definition of anomaly since this paper only takes low-possibility motion patterns as anomalies.

The results are illustrated in Figure 3. (a), (b) is the frame-level and pixel-level ROC curve, respectively. It can be seen that our algorithm performs comparable with state-of-art methods at frame-level and outperforms state-of-art methods at pixel-level when the false positive rate is less than around 24%. This improvement is important because it is not practical to tolerate high false alarm rate. And (c) is the run-


**Fig. 2**. Detection results of real-life surveillance video. Group (a) shows two detected anomalies of one scene: the pedestrian crossing the road at wrong location and a car entering the side-road at wrong location. Group (b) shows two detected reversely running motorcycles of another scene. Group (c) shows two anomalies of different scenes: detected trunks running on the wrong side of road at night and the pedestrian across the road at wrong location.




Running Time Comparison on UCSD Ped1 Dataset

| Methods | [2] | [4] | [6] | **Ours** |
|---|---|---|---|---|
| Sec/Frame | 25 | 3.8 | 0.0070 | **0.0326** |
| FPS | 0.04 | 0.26 | 142.86 | **30.67** |
| Platform | - | - | Matlab | C++ |
| CPU(GHz) | 3.0 | 2.6 | 3.4 | 2.5 |
| Mem(GB) | 2.0 | 2.0 | 8.0 | 8.0 |

(c)

**Fig. 3**. Comparison of ROC curves on UCSD ped1 dataset. Here Adam refers to [1]; MDT refers to [3]; social force refers to [14]; sparse and sparse combination refer to [1] and [6], respectively. (a) is the frame-level ROC curve; Note that frame-level AUC is not the bigger the better if the pixel-level AUC is not in accord with it. (b) is the pixel-level comparison. (c) is a table of running time comparison.

ning time comparison. Our algorithm runs real-time on this benchmark dataset, which is much faster than existing algorithms [3, 1]. Algorithm [6] reaches an impressing speed by resizing frames to $30 \cdot 30$ and other small resolutions. However, this will not work if the abnormal objects in original frame are small.

## 4. CONCLUSION

We propose an efficient algorithm for detecting anomaly in traffic surveillance video based on robust sparse optical flow. The sparse optical flow computation is improved in both robustness and efficiency. A foreground channel is added to HOF feature to detect long-term static objects. The algorithm is validated on real-life traffic surveillance and a benchmark dataset to prove its effectiveness. The algorithm runs real-time and is hundreds of times faster than a number of comparative algorithms. The speed gain is achieved by a fast feature extraction design and a simple detection model. This research was partially supported by the National Natural Science Foundation of China under Grant 61403403 and 61405252.

## 5. REFERENCES


[1] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 3449–3456, 2011.

[2] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, pp. 555–560, 2008.

[3] V. Mahadevan, W. Li, V. Bhalodia, , and N. Vasconcelos, "Anomaly detection in crowded scenes," *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 1975–1981, 2010.

[4] P. Jodoin, V. Saligrama, and J. Konrad, "Behavior subtraction," *PROCEEDINGS-SPIE THE INTERNATIONAL SOCIETY FOR OPTICAL ENGINEERING*, p. 6822, 2008.

[5] T. WANG, H. SNOUSSI, , and F. SMACH, "Detection of visual abnormal events via one-class svm," *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, pp. 113–119, 2012.

[6] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 fps in matlab," *Computer Vision (ICCV), IEEE International Conference on*, pp. 2720–2727, 2013.

[7] P. Jodoin, V. Saligrama, and J. Konrad, "Behavior subtraction," *Image Processing, IEEE Transactions on*, pp. 4244–4255, 2012.

[8] V. Saligrama and Z. Chen, "Video anomaly detection based on local statistical aggregates," *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 2112–2119, 2012.

[9] T. Dean, M.A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik, "Fast, accurate detection of 100,000 object classes on a single machine," *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2013.

[10] A. Basharat, A. Gritai, and M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 1–8, 2008.

[11] B.D. Lucas and Kanade, "An iterative image registration technique with an application to stereo vision," *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 674–679, 1981.

[12] C. Tomasi and J. Shi, "Good features to track," *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 593–593, 1994.

[13] Z. Kalal, "Tracking learning detection," *PhD Thesis, University of Surrey*, vol. 30, pp. 555–560, January 2011.

[14] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 935–942, 2009.

[15] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Ucsd ped dataset," http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm, 2013.