MVA2009 IAPR Conference on Machine Vision Applications, May 20-22, 2009, Yokohama, JAPAN

Unsupervised Abnormal Behavior Detection for Real-time Surveillance Using Observed History

Tsz-Ho Yu
Dept. of Computer Science & Engineering
The Chinese University of Hong Kong
thyu@cse.cuhk.edu.hk

Yiu-Sang Moon
Dept. of Computer Science & Engineering
The Chinese University of Hong Kong
ysmoon@cse.cuhk.edu.hk

Abstract

This paper presents a novel method of utilizing observed history for detecting abnormal behaviors in surveillance applications. An unsupervised algorithm is proposed to detect abnormal behaviors and re-train itself in real-time. Motion vectors of objects are estimated using the optical flow method. Encoded feature vectors are stored in an observation matrix, abnormal behaviors can be detected by applying principal copmonent analysis (PCA) on the matrix. This method has been evaluated under both indoor and outdoor surveillance scenarios. It demonstrates promising results that this detection procedure is able to discover abnormal behaviors and adapt to changes in the behavioral patterns incrementally.

1 Introduction

Discovering abnormal behaviors is the key step for many computer vision applications, especially for smart visual surveillance. Although human action recognition has attracted much attention recently, some issues remain largely unsolved for a deployable surveillance system. First, abnormal behaviors are difficult to be defined formally[9]. A suspicious behavior in one scene can be regarded as normal in another environments. In spite of this, many of the current approaches rely on supervised learning methods, the "abnormal behaviors" are required to be well-defined in a labeled data set for training. These approaches are not feasible in realistic surveillance, when the definition of "unusual behaviors" changes with the environments. In addition, some algorithms rely on detection of local features, such as the shape of object contours [7, 8]. These approaches acheieve high performances when foreground objects can be extracted accurately. However, the background subtraction algorithm is affected by occlusions or clutter background. Moreover, many techniques on unusual behavior detection cannot perform in real-time, as they require the complete video for modeling the abnormal behaviors [9, 7, 3]. Hence, these methods cannot respond immediately to control the tracking cameras, or to notify the human operator about the abnormal behavior detected.

The major contribution of our work is to design a method for detecting unusual behavior in real-time. Viewing the problem from a different perspective, the proposed method emphasizes on using observed history as reference data. Without using a fixed labeled data set, this method is capable to update itself incrementally, adapting to changes of behavioral patterns in the surveillance environment.

The rest of the paper is structured as follows: Section 2 explains the feature extraction procedure using pyramidal Lucas-Kanade algorithm. Section 3 describes the abnormal behavior detection and localization methodologies. Experiments have been performed to justify the feasibility of our proposed method; the results are reported in section 4. Finally, section 5 outlines our conclusion.

2 Feature Extraction

2.1 Motion Detection Using Optical Flow

Surveillance videos captured from the cameras are first preprocessed to undergo foregorund-background segmentation. Foreground pixels are extracted by applying the *Mixture of Gaussian* method[6]. Subsequently, isolated foreground and noise pixels are deleted by morphological opening operators[2]. Features of objects' movements are extracted by computing the dense optical flow field from the video. The optical flow approach is employed because it only requires two consecutive frames for estimating the motion, thus new features can be extracted from every frame. As described in figure 1, the feature points in background regions are masked by the foreground pixel mask, such that the noises introduced by flow vectors on background can be reduced.

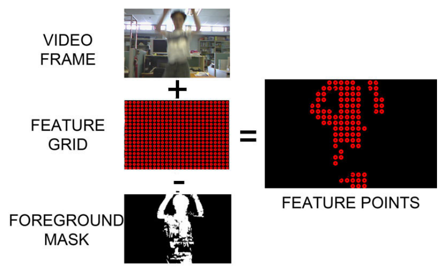

Figure 1: Feature points of optical flow are masked by the foreground regions.

Since traditional *Lucas-Kanade method*[4] assumes small and constant movements, it is not feasible in detecting fast movements. Addressing this problem, the pyramidal version of Lucas-Kanade method[1] is adopted. An image pyramid is constructed, such that larger flow vectors can be estimated more accurately. Figure 2 shows some examples of flow vectors extracted from the videos.

2.2 Running Flow Histograms

As a result of the optical flow algorithm, flow vectors are obtained from the masked feature points. The set of all J flow vectors in the i-th frame F_i is represented in (1):

$$\begin{aligned} F_i &= \{f_{i1}, f_{i2}, ..., f_{ij}, ..., f_{iJ}\} \\ f_{ij} &= [X_{ij}, Y_{ij}, \theta_{ij}, S_{ij}], \end{aligned} \quad (1)$$

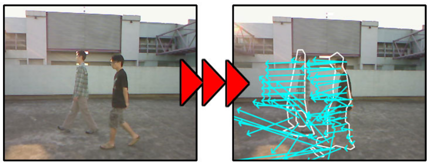

Figure 2: Flow vectors obtained from the pyramidal Lucas-Kanade algorithm

where the coordinate (X_{ij},Y_{ij}) denotes the location of the feature point; θ_{ij} and S_{ij} represent the orientation and veocity of the corresponding flow vector respectively. A two dimensional flow histogram is constructed for each frame, with respect to the values of θ and S of the flow vectors. The vectors in the same frame are grouped into $N_v \times N_\theta$ bins, according to the velocity and orientation respectively. A flow histogram summarizes the dynamic characteristics in the scene at a particular moment. Figure 3 visualizes the structure of a flow histogram.

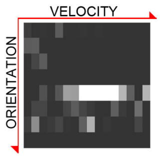

Figure 3: The structure of a flow histogram

In the proposed method, an event is represented by integrating individual flow histograms from the observed history. Assuming $0 \leq v < N_v$ and $0 \leq \theta < N_\theta$, the *Running Flow Histogram*(RFH) of the i-th frame $h'_i(\theta, v)$ is calculated in (3).

$$h'_{ni}(\theta, v) = Normalize(h'_i(\theta, v)) \tag{2}$$
$$h''_i(\theta, v) = \alpha_h \times h'_{ni}(\theta, v) + (1 - \alpha_h) \times h''_{i-1}(\theta, v) \tag{3}$$

A running flow histogram is actually the weighted running average of the normalized flow histogram from previous observations. The learning rate parameter α_h controls the length of the event represented by the running flow histogram. When a frame is captured from the security camera, a new RFH is computed from the current histogram and the RFH in the previous frame. After that, for analyzing the recorded events by PCA in the learning process, each RFH is reshaped into a feature vector $h(\theta * N_v + v)$, as shown in (4).

$$h_i(\theta * N_v + v) = h''_i(\theta, v). \tag{4}$$

The feasibility of using flow histograms in representing scene features is justified in figure 4: Similar events in the scene demonstrate high correlations in their RFHs; on the contrary, a large difference between two RFHs implies the events are not similar.

3 Unsupervised Detection

As discussed in section 1, the main characteristic of our method is to detect abnormal behaviors by referencing the observed history, instead of training a classifier from a label training set. The underlying concept is based on the correlations between similar events. Usually, normal behaviors occur frequently, hence a large amount of similar

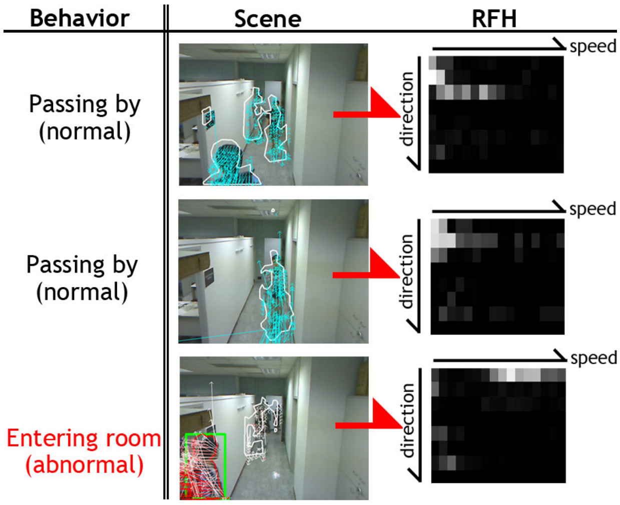

Figure 4: Top row and middle row: the scenes and their corresponding RFHs of similar behaviors. Bottom row: the scene and RFH of an abnormal behavior. Interestingly, similar events possess high correlations between their RFHs

RFHs are recorded. Consequently, the distributions of the observed RFHs concentrate over several specific ranges. On the contrary, abnormal behaviors are rare; thus the RFHs of abnormal events are scattered and isolated from the major distributions. Leveraging the high correlations between normal behaviors, frames with abnormal behaviors can be discovered by detecting the outliers from a set of previously observed RFHs. A behavior is classified as abnormal if it shows a large deviation from the recorded events in the history. To facilitate the unsupervised detection process, an observation matrix is used to store the RFHs of the observed history.

3.1 Observation Matrix

The observation matrix is implemented using a circular linked-list; the structure of this linked-list is illustrated in Figure 5. Assume the observation linked-list contains at most L entries, the observation matrix H has a size of $L \times N_d * N_v$. The matrix is initialized by the RFHs computed from the first L frames, as described in (5).

$$H = [\ h_1 \quad h_2 \quad h_3 \quad \ldots \quad h_L\ ]. \tag{5}$$

The list adopts a *"first in, first out"* (FIFO) updating policy: When a new frame is acquired from the camera, its RFH replaces the oldest column in the behavior matrix. In a long run, old observations are gradually replaced by the more recent ones; the feature vectors inside the matrix change incrementally. Since the detection criteria is detemrined by the observed history stored in the observation matrix, the classifier can be adjusted dynamically.

3.2 Detecting Abnormal Behavior

Eigenvectors and eigenvalues of the observation matrix are computed using PCA[5]. This detection process attempts to reconstruct the information of the current frame using the eigenvectors of the observation matrix. Abnormal behaviors can be detected by measuring the completeness of the reconstruction process. For the mean-subtracted observed matrix $\overline{H}$, the covariance matrix C is computed

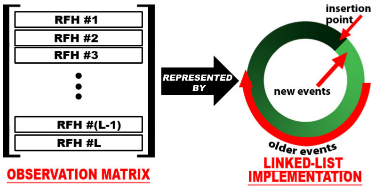

Figure 5: An observation matrix implemented by a circular linked-buffer

using the outer product of $\overline{H}$. Afterwards, the matrix of eigenvectors V and diagonal matrix of eigenvalues E can be calculated by solving (6):

$$V^{-1}CV = E. \tag{6}$$

The eigenvectors are sorted by their corresponding eigenvalues in descending order. The first L' eigenvectors are selected to a matrix V' for abnormal behavior detection. Subsequently, the current RFH is reconstructed into h_i^r by eigen projections in (7).

$$\begin{aligned} &\text{Projection:} && \mathbf{h}_i = (h_i - \overline{H}) * V'^T \\ &\text{Back-projection:} && h_i^r = \mathbf{h}_i * V' + \overline{H}. \end{aligned} \tag{7}$$

Abnormal behaviors are distinguished by measuring how much information can be recovered by using the principal components obtained from the observation matrix. Since an unusual behavior does not have high correlations with the normal behaviors in the history, its RFH cannot be reconstructed easily. By leveraging this property, the recovered energy ratio E_R indicates the presence of unusual behavior in the current frame, as defined in (8):

$$E_R = \frac{\sum_{n=0}^{N_\theta * N_v - 1} (h_i^r[n])^2}{\sum_{n=0}^{N_\theta * N_v - 1} (h_i[n])^2} \times 100\% \tag{8}$$

From (9), abnormal behavior(s) is detected in a frame when the recovery energy ratio E_R is smaller than a pre-defined threshold $thresh_r 1$. This threshold determines the overall sensitivity of the detection process.

$$E_R \leq thresh_{r1} \tag{9}$$

3.3 Abnormal Behavior Localization

Abnormal behaviors are localized by analyzing back-projected RFH. For the i-th frame, the absolute difference between the original and back-projected RFHs is computed. Weighted factors W_i assigned to each of the bins, as described in (10).

$$W_i(\theta * N_v + v) = |h_i^r(\theta * N_v + v) - h_i(\theta * N_v + v)|. \tag{10}$$

W_i represents the anomaly of a flow vector at the i-th frame, a large value in W_i indicates that flow vectors cannot be recovered by PCA efficiently. The anomaly weights $W_i(d * N_v + v)$ are assigned to all flow vectors in the scene, according to its corresponding bin in h_i. Blob analysis is performed to detect the connected components $[B_0 ... B_{ji} ... B_{Mi}]$ from the foreground regions. For each connected component, its average "anomally" A can be calculated from the sum of all weights that assigned to the vectors over the region's area.

$$A(B) = \frac{\sum_{F \in B} W_i}{AREA(B)}. \tag{11}$$

Hence, an object's behavior can be classified as abnormal if it's average anomally is greater than the threshold $thresh_{r2}$:

$$A(B) \geq thresh_{r2} \tag{12}$$

4 Evaluation

4.1 Experiment Setup

To justify the effectiveness of the proposed approach, several experiments were conducted in real-life surveillance conditions. Security cameras were installed in three different surveillance scenarios. The first security camera was mounted in an outdoor environment, so as to evaluate the performance under varying illumination. The second security camera was installed on at the end of a corridor, simulating the arrangement of a realistic closed-circuit television (CCTV) system. At one side of the corridor, there was a control room; whereas most of the people passed by the corridor, only a few people entered the room. The third security was mounted in the laboratory; fast abnormal actions were performed in front of the camera, in order to evaluate the response time of the proposed system. Abnormal events were counted manually in the surveillance videos, a successful detection was defined by the action that was detected by both the proposed method and manual observation. The number of false positive detection were also recorded.

4.2 Evaluation Results

The evaluation results are summarized in Table 1. The proposed detection method shows promising results. Without using any explicit training data, our method detects abnormal events automatically with a high detection rate. In addition, the detection algorithm is capable to perform in real-time and issue timely responses when abnormal behaviors are detected. Moreover, the detection algorithm is able to learn incrementally; the classification criteria can be adjusted according to changes in the surveillance environment. The proposed algorithm is robust against occlusions and segmentation errors. However, some of the false positive detections were recorded. Most of the false positive cases were produced at the beginning of the operation, when not enough examples were learned from the history. Figure 6 shows some of the detection results obtained from the experiments.

5 Conclusion

An innovative method for abnormal behavior detection is presented in this paper. We have designed a unsupervised learning method for discovering abnormal movements for visual surveillance applications. Without using a labeled data set, the detector updates itself incrementally by learning the behavioral patterns from previous observations. Applying PCA on the observation matrix, this method is able to detect abnormal behaviors in real-time. The evaluation results are promising: it is able to detect abnormal behaviors accurately without any training data; it also updates itself incrementally and conforms to the changes in environment.

Scenario	Length	No. of Abnormal Behaviors	Successful Detection	Detection Rate (Detected/No. of Abnormal Behavior)	No. of False Positives
Outdoor	333 seconds @ 15fps	11	10	90.9%	4
Corridor	333 seconds @ 30fps	17	16	94.1%	6
Laboratory	150 seconds @ 30fps	7	6	85.7%	2

Table 1: Experimental results of the proposed abnormal behavior detection algorithm

Figure 6: Sample detection results

Acknowledgement

The work described in this paper was substantially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. 415207).

References

[1] J. Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker: Description of the algorithm, 2002.

[2] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[3] Y. Li, C. Xu, J. Liu, and X. Tang. Detecting irregularity in videos using kernel estimation and kd trees. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 639–642, New York, NY, USA, 2006. ACM.

[4] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI81*, pages 674–679, 1981.

[5] J. Shlens. A tutorial on principal component analysis, December 2005.

[6] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:2246, 1999.

[7] X. Wu, Y. Ou, H. Qian, and Y. Xu. A detection system for human abnormal behavior. *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1204–1208, Aug. 2005.

[8] T. Xiang and S. Gong. Video behavior profiling for anomaly detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(5):893–908, May 2008.

[9] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2:II–819–II–826 Vol.2, June-2 July 2004.