

AUTOMATIC SUSPICIOUS BEHAVIOR DETECTION FROM A SMALL BOOTSTRAP SET

Kan Ouivirach, Shashi Gharti and Matthew N. Dailey

School of Engineering and Technology, Asian Institute of Technology, Bangkok, Thailand

Keywords: Hidden Markov Models, Behavior Clustering, Anomaly Detection, Bootstrapping.

Abstract: We propose and evaluate a new method for automatic identification of suspicious behavior in video surveillance data. It partitions the bootstrap set into clusters then assigns new observation sequences to clusters based on statistical tests of HMM log likelihood scores. In an evaluation on a real-world testbed video surveillance data set, the method achieves a false alarm rate of 7.4% at a 100% hit rate. It is thus a practical and effective solution to the problem of inducing scene-specific statistical models useful for bringing suspicious behavior to the attention of human security personnel.

1 INTRODUCTION

We focus on enhancing security by performing intelligent filtering of typical events and automatically bringing suspicious events to the attention of human security personnel.

Pre-trained hidden Markov model (HMMs) and other dynamic Bayesian networks such as conditional random fields (CRFs) have been widely used in this area (Gao et al., 2004; Sminchisescu et al., 2005). Some of the existing work relies on having a priori known behavior classes (Nair and Clark, 2002; Wu et al., 2005).

More recent work uses unsupervised analysis and clustering of behaviors (Li et al., 2006; Swears et al., 2008). Xiang and Gong (2005) model the distribution of activity data in a scene using a Gaussian mixture model (GMM) and employ the Bayesian information criterion (BIC) to select the optimal number of behavior classes prior to HMM training.

We propose to use HMM-based clustering on a small bootstrap set of sequences labeled as normal or suspicious. After bootstrapping is complete, we assign new observation sequences to behavior clusters using statistical tests on the log likelihood of the sequence according to the corresponding HMMs. The cluster-specific likelihood threshold is learned rather than set arbitrarily.

In this paper, we briefly describe our method. For more details, see the full version of this paper.¹

¹<http://www.cs.ait.ac.th/techreports/AIT-CSIM-TR-2012-1.pdf>.

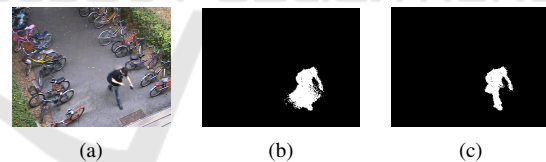


Figure 1: Sample foreground extraction and shadow removal results. (a) Original image. (b) Foreground pixels according to background model. (c) Foreground pixels after shadow removal.

2 BEHAVIOR MODEL BOOTSTRAPPING

To build the bootstrap set, over a short period such as one week, we first discard no-motion frames. We then use the background modeling technique proposed by Poppe et al. (2007). We use normalized cross correlation (NCC) to eliminate shadows cast by moving objects. Sample results from the foreground extraction and shadow removal procedures are shown in Figure 1. We apply morphological opening then closing operations to obtain the connected components, then we filter out any components whose size is below threshold. We finally represent each blob (connected foreground component) i at time t by a feature vector f_i^t containing the blob's centroid, size, and aspect ratio, a unit-normalized motion vector for the blob compared to the previous frame, and the blob's speed. Next we map each feature vector f_i^t to a discrete category (cluster ID) in the set $V = \{v_1, v_2, \dots, v_U\}$, where

U is the number of categories, using k -means clustering. A behavior sequence is finally represented as a sequence of cluster IDs.

In blob tracking, we associate blobs in the current frame with the tracks from the previous frame using bounding box overlap rule. Blobs corresponding to isolated moving objects are associated with unique tracks. When no merge or split occurs, each blob either matches one of the existing tracks or is classified as new, in which case a new track is created. We destroy tracks that are inactive for some number of frames. We use the color coherence vector (CCV) (Pass et al., 1996) as an appearance model to handle cases of blob merges and splits. When tracks merge, we group them, but keep their identities separate, and when tracks split, we attempt to associate the new blobs with the correct tracks or groups of tracks by comparing their CCVs. This works well on typical simple cases such as those as shown in Figure 2, but it can make mistakes with more complex cases.

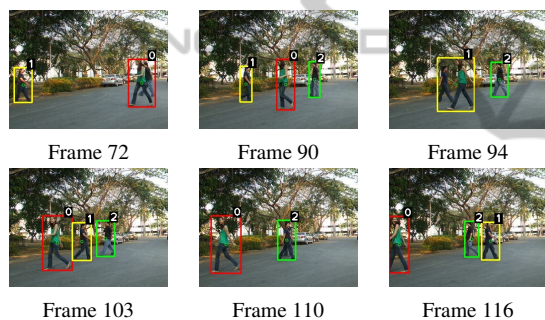


Figure 2: Sample blob tracking results for typical simple case.

After blob tracking, we obtain, from a given video, a set of observation sequences describing the motion and appearance of every distinguishable moving object in the scene. We next partition the observation sequences into clusters of similar behaviors then model the sequences within each cluster using a simple linear HMM. We use the method from our previous work (Ouvirach and Dailey, 2010), which first uses dynamic time warping (DTW) to obtain a distance matrix for the set of observation sequences then performs agglomerative hierarchical clustering on the distance matrix to obtain a dendrogram.

To determine where to cut off the dendrogram, we traverse the DTW dendrogram in depth-first order from the root and attempt to model the observation sequences within the corresponding subtree using a single linear HMM. If, after training, the HMM is unable to “explain” (in the sense described below) the sequences associated with the current subtree, we discard the HMM then recursively attempt to model each

of the current node’s children. Whenever the HMM is able to explain the observation sequences associated with the current node’s subtree, we retain the HMM and prune the tree.

A HMM is said to explain a cluster c if there are no more than N_c sequences in cluster c whose per-observation log-likelihood is less than a threshold p_c . To determine the optimal rejection threshold p_c for cluster c , we use an approach similar to that of Oates et al. (2001). We generate random sequences from the HMM and then calculate the mean μ_c and standard deviation σ_c of the per-observation log likelihood over the set of generated sequences. After obtaining the statistics of the per-observation log likelihood, we let p_c be $\mu_c - z\sigma_c$, where z is an experimentally tuned parameter.

3 ANOMALY DETECTION

For anomaly detection, we propose a *semi-supervised* method that self-calibrates itself from the bootstrap set. We apply the algorithm of Section 2 to *both the positive and negative sequences* in the bootstrap set. We identify each cluster as a “normal” cluster if *all* of the sequences falling into it are labeled as normal, or identify it as an “abnormal” cluster if *any* of the sequences falling into it are labeled as abnormal. New sequences are classified as normal if the most likely HMM for the input sequence is associated with a cluster of normal sequences *and* the z -scaled per-observation log likelihood of the sequence under that most likely model is greater than a global empirically determined threshold z .

4 EXPERIMENTAL RESULTS

We recorded video from the scene in front of a building for one week. We labeled 625 occurrences of walking into the building, walking out, riding bicycles in, and riding bicycles out as normal and the remaining 35 occurrences as suspicious or abnormal.

4.1 Model Configuration Selection

Towards model identification, we performed a series of experiments with different bootstrap parameter settings and selected the configuration with the highest accuracy in separating the normal sequences from the abnormal sequences on the bootstrap sequence set, as measured by the false positive rate for abnormal sequences. Every bootstrap cluster containing an abnormal sequence is considered abnormal, so we al-

Table 1: Anomaly detection results for the proposed method, k -NN, PCA, and SVM in Experiment I, II, III, and IV, respectively. For PCA, we include 11 abnormal sequences from the bootstrap set in the test set, so the total number of positives is 35.

Method	TP	FP	TN	FN	TPR	FPR
Ours	24	36	450	0	1	0.074
1-NN	19	1	485	5	0.792	0.002
PCA	35	421	65	0	1	0.87
SVM	24	228	258	0	1	0.469

ways obtain 100% detection on the bootstrap set; the only discriminating factor is the false positive rate. To find the distribution (parameters μ_c and σ_c) of the per-observation log likelihood for a particular HMM, we always generated 1000 sequences of 150 observations then used a z -threshold of 2.0. We fixed the parameter N_c (the number of deviant patterns allowed in a cluster) to 10.

Based on the false positive rate criterion described above, we selected the model configuration consisting of HMMs with five states and seven tokens trained on 150 bootstrap sequences.

4.2 Anomaly Detection

Here we describe four experiments to evaluate our anomaly detection method. For each method, we use the 150-sequence bootstrap sequence set from Section 4.1 as training data and test on the remaining 510 sequences. In every experiment, we calculate an ROC curve and select the detection threshold yielding the best false positive rate at a 100% hit rate. Our experimental hypothesis was that the proposed method for modeling scene-specific behavior patterns should obtain better false positive rates than the traditional methods.

In Experiment I, the red line in Figure 3 represents the ROC curve for our method as we vary the likelihood threshold at which a sequence is considered anomalous. Note that the ROC does not intersect the point (0,0) because any sequence that is most likely under one of the HMMs modeling anomalous sequences in the bootstrap set is automatically classified as anomalous regardless of the threshold. Table 1 shows the detailed performance of this model, and Figure 4 shows an example of a sequence classified as abnormal.

In Experiment II, we applied k -nearest neighbors (k -NN) using the same division of sequences into training and testing as in Experiment I. As the distance measure, we used the same DTW measure we used for hierarchical clustering of the bootstrap patterns in our method. We varied k from 1 to 5. The best result for k -NN is shown in Table 1. While the false

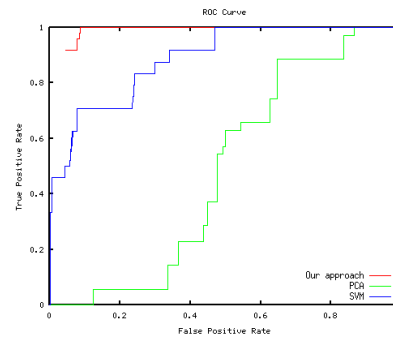


Figure 3: Anomaly detection ROC curves. Red, green and blue lines represent ROCs for the proposed method in Experiment I, PCA-based anomaly detection in Experiment III, and SVM-based anomaly detection in Experiment IV, respectively.

positive rates are much lower than those obtained in our method, the hit rates are unacceptable.

In Experiment III, we classified sequences as normal or anomalous using a Gaussian density estimator derived from principal components analysis (PCA). We calculated, for each sequence in the testbed data set, a summary vector consisting of the means and standard deviations of each observation vector element over the entire sequence. With seven features in the observation vector, we obtained a 14-element vector summarizing each the sequence. After feature summarization, we normalized each component of the summary vector by z -scaling. Then, since we are performing probability density estimation for the normal patterns, we applied PCA to the 139 normal sequences in the bootstrap set. We chose the number of principal components accounting for 80% of the variance in the bootstrap data. Finally, we classified the remaining 521 test sequences using the PCA model to calculate the Mahalanobis distance of each sequence’s summary vector to the mean of the normal bootstrap patterns’ summary vectors. The green line in Figure 3 is the ROC curve obtained by varying the Mahalanobis distance threshold, and Table 1 shows detailed results for anomaly detection at a 100% hit rate. The high false positive rate at this threshold and the overall poor performance in the ROC analysis show that PCA is clearly inferior to our proposed method.

In Experiment IV, here we used the same summary vector technique used in Experiment III but performed supervised classification using support vector machines. We used the radial basis function kernel implementation in LIBSVM (Chang and Lin, 2001) with grid search for the optimal hyperparameters using five-fold cross validation on the training set (150 sequences). The blue line in Figure 3 is the ROC curve obtained by varying the threshold on the signed

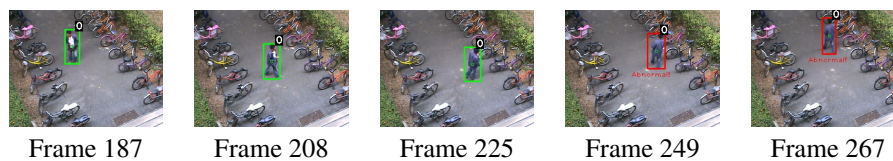


Figure 4: Example anomaly detected by the proposed method in Experiment I. The sequence contains a person walking around looking for an unlocked bicycle.

distance to the separating hyperplane used for classification as normal or abnormal. Table 1 shows the detailed anomaly detection results for SVMs at a 100% hit rate. Although the results are clearly better than those obtained from k -NN or PCA, they are also clearly inferior to those obtained in Experiment I.

5 DISCUSSION AND CONCLUSIONS

We have proposed and evaluated a new method for bootstrapping scene-specific anomalous human behavior detection systems. It requires minimal involvement of a human operator; the only required action is to label the patterns in a small bootstrap set as normal or anomalous. With a bootstrap set of 150 sequences, the method achieves a false positive rate of merely 7.4% at a hit rate of 100%. The experiments demonstrate that with a collection of simple HMMs, it is possible to learn a complex set of varied behaviors occurring in a specific scene. Deploying our system on a large video sensor network would potentially lead to substantial increases in the productivity of human monitors.

The main limitation of our current method is that the blob tracking process is not robust for complex events involving multiple people. The method also does not allow evolution of the learned bootstrap model over time. In future work, we plan to address these limitations.

ACKNOWLEDGEMENTS

This work was partly supported by a grant from the Royal Thai Government to MND and by graduate fellowships from the Royal Thai Government to KO. We thank the AIT Computer Vision Group for valuable discussions and comments on this work.

REFERENCES

- Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Gao, J., Hauptmann, A. G., Bharucha, A., and Wactlar, H. D. (2004). Dining activity analysis using a hidden Markov model. In *International Conference on Pattern Recognition (ICPR)*, pages 915–918.
- Li, H., Hu, Z., Wu, Y., and Wu, F. (2006). Behavior modeling and recognition based on space-time image features. In *International Conference on Pattern Recognition (ICPR)*, pages 243–246.
- Nair, V. and Clark, J. (2002). Automated visual surveillance using hidden Markov models. In *Vision Interface Conference*, pages 88–92.
- Oates, T., Firoiu, L., and Cohen, P. R. (2001). Using dynamic time warping to bootstrap HMM-based clustering of time series. In *Sequence Learning: Paradigms, Algorithms, and Applications*, pages 5–52.
- Ouivirach, K. and Dailey, M. N. (2010). Clustering human behaviors with dynamic time warping and hidden Markov models for a video surveillance system. In *ECTI-CON*, pages 884–888.
- Pass, G., Zabih, R., and Miller, J. (1996). Comparing images using color coherence vectors. In *International Conference on Multimedia*, pages 65–73.
- Poppe, C., Martens, G., Lambert, P., and de Walle, R. V. (2007). Improved background mixture models for video surveillance applications. In *Asian Conference on Computer Vision (ACCV)*, pages 251–260.
- Sminchisescu, C., Kanaujia, A., Li, Z., and Metaxas, D. (2005). Conditional random fields for contextual human motion recognition. In *International Conference on Computer Vision (ICCV)*, pages 1808–1815.
- Swears, E., Hoogs, A., and Perera, A. (2008). Learning motion patterns in surveillance video using HMM clustering. In *IEEE Workshop on Motion and video Computing (WMVC)*, pages 1–8.
- Wu, X., Ou, Y., Qian, H., and Xu, Y. (2005). A detection system for human abnormal behavior. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1204–1208.
- Xiang, T. and Gong, S. (2005). Video behaviour profiling and abnormality detection without manual labelling. In *International Conference on Computer Vision (ICCV)*, pages 1238–1245.