Video Behavior Profiling for Anomaly Detection

Tao Xiang, *Member*, *IEEE*, and Shaogang Gong

Abstract—This paper aims to address the problem of modeling video behavior captured in surveillance videos for the applications of online normal behavior recognition and anomaly detection. A novel framework is developed for automatic behavior profiling and online anomaly sampling/detection without any manual labeling of the training data set. The framework consists of the following key components: 1) A compact and effective behavior representation method is developed based on discrete-scene event detection. The similarity between behavior patterns are measured based on modeling each pattern using a Dynamic Bayesian Network (DBN). 2) The natural grouping of behavior patterns is discovered through a novel spectral clustering algorithm with unsupervised model selection and feature selection on the eigenvectors of a normalized affinity matrix. 3) A composite generative behavior model is constructed that is capable of generalizing from a small training set to accommodate variations in unseen normal behavior patterns. 4) A runtime accumulative anomaly measure is introduced to detect abnormal behavior, whereas normal behavior patterns are recognized when sufficient visual evidence has become available based on an online Likelihood Ratio Test (LRT) method. This ensures robust and reliable anomaly detection and normal behavior recognition at the shortest possible time. The effectiveness and robustness of our approach is demonstrated through experiments using noisy and sparse data sets collected from both indoor and outdoor surveillance scenarios. In particular, it is shown that a behavior model trained using an unlabeled data set is superior to those trained using the same but labeled data set in detecting anomaly from an unseen video. The experiments also suggest that our online LRT-based behavior recognition approach is advantageous over the commonly used *Maximum Likelihood* (ML) method in differentiating ambiguities among different behavior classes observed online.

Index Terms—Behavior profiling, anomaly detection, dynamic scene modeling, spectral clustering, feature selection, Dynamic Bayesian Networks.

1 Introduction

There is an increasing demand for automatic methods for analyzing the vast quantities of surveillance video data generated continuously by closed-circuit television (CCTV) systems. One of the key objectives of deploying an automated visual surveillance system is to detect abnormal behavior patterns and recognize the normal ones. To achieve this objective, previously observed behavior patterns need to be analyzed and profiled, upon which a criterion on what is normal/abnormal is drawn and applied to newly captured patterns for anomaly detection. Due to the large amount of surveillance video data to be analyzed and the real-time nature of many surveillance applications, it is very desirable to have an automated system that runs in real time and requires little human intervention. In the paper, we aim to develop such a system that is based on fully unsupervised behavior profiling and robust online anomaly detection.

Let us first define the problem of automatic behavior profiling for anomaly detection. Given a 24/7 continuously recorded video or an online CCTV input, the goal of automatic behavior profiling is to learn a model that is capable of detecting unseen abnormal behavior patterns while recognizing novel instances of expected normal behavior patterns. In this context, we define an anomaly as an atypical behavior pattern that is not represented by sufficient samples in a training data set but critically satisfies the specificity constraint to an abnormal pattern. This is because one of the main challenges for the model is to differentiate anomaly from outliers caused by noisy visual features used for behavior representation. The effectiveness of a behavior profiling algorithm shall be measured by 1) how well anomalies can be detected (that is, measuring specificity to expected patterns of behavior) and 2) how accurately and robustly different classes of normal behavior patterns can be recognized (that is, maximizing between-class discrimination).

To solve the problem, we develop a novel framework for fully unsupervised behavior profiling and online anomaly detection. Our framework has the following key components:

1. *A scene event-based behavior representation*. Due to the space-time nature of behavior patterns and their variable durations, we need to develop a compact and effective behavior representation scheme and to deal with time warping. We adopt a discrete scene event-based image feature extraction approach [8]. This is different from most previous approaches such as [24], [16], [14], [3] where image features are extracted based on object tracking. A discrete event-based behavior representation aims to avoid the difficulties associated with tracking under occlusion in noisy scenes [8]. Each behavior pattern is modeled using a Dynamic Bayesian Network (DBN) [7], which provides a suitable means for time warping and measuring the affinity between behavior patterns.
2. *Behavior profiling based on discovering the natural grouping of behavior patterns using the relevant eigenvectors of a normalized behavior affinity matrix*. A number of affinity matrix-based clustering techniques have been proposed recently [25], [23], [30]. However, these approaches require a known number of clusters. Given an unlabeled data set, the number of behavior classes is unknown in our case. To automatically

• *The authors are with the Department of Computer Science, Queen Mary, University of London, Mile End Road, London E1 4NS, UK. E-mail: {txiang, sgg}@dcs.qmul.ac.uk.*

Manuscript received 17 Feb. 2006; revised 20 Oct. 2006; accepted 11 June 2007; published online 25 June 2007.
Recommended for acceptance by H. Sawhney.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0168-0206.
Digital Object Identifier no. 10.1109/TPAMI.2007.70731.

determine the number of clusters, we propose to first perform unsupervised feature selection to eliminate those eigenvectors that are irrelevant/uninformative in behavior pattern grouping. To this end, a novel feature selection algorithm is derived, which makes use of the a priori knowledge on the relevance of each eigenvector. Our unsupervised feature selection algorithm differs from the existing techniques such as [12], [6] in that it is simpler, more robust, and thus able to work more effectively even with sparse and noisy data.

3. *A composite generative behavior model using a mixture of DBNs*. The advantages of such a generative behavior model are twofold: a) It can accommodate well the variations in the unseen normal behavior patterns in terms of both duration and temporal ordering by generalizing from a training set of a limited number of samples. This is important because in reality, the same normal behavior can be executed in many different normal ways. These variations cannot possibly be captured in a limited training data set and need to be dealt with by a learned behavior model. b) Such a model is robust to errors in behavior representation. A mixture of DBNs can cope with errors that occurred at individual frames and is also able to distinguish an error corrupted normal behavior pattern from an abnormal one.
4. *Online anomaly detection using a runtime accumulative anomaly measure and normal behavior recognition using an online Likelihood Ratio Test (LRT) method*. A runtime accumulative measure is introduced to determine how normal/abnormal an unseen behavior pattern is on the fly. The behavior pattern is then recognized as one of the normal behavior classes if detected as being normal. Normal behavior recognition is carried out using an online LRT method, which holds the decision on recognition until sufficient visual features have become available. This is in order to overcome any ambiguity among different behavior classes observed online due to insufficient visual evidence at a given time instance. By doing so, robust behavior recognition and anomaly detection are ensured at the shortest possible time, as opposed to previous work such as [2], [8], [16], which requires completed behavior patterns to be observed. Our online LRT-based behavior recognition approach is also advantageous over previous ones based on the *Maximum Likelihood* (ML) method [31], [8], [16]. An ML-based approach makes a forced decision on behavior recognition at each time instance without considering the reliability and sufficiency of the accumulated visual evidence. Consequently, it can be error prone.

Note that our framework is fully unsupervised in that manual data labeling is avoided in both the feature extraction for behavior representation and the discovery of the natural grouping of behavior patterns. There are a number of motivations for performing behavior profiling using unlabeled data: First, manual labeling of behavior patterns is laborious and often rendered impractical given the vast amount of surveillance video data to be processed. More critically though, manual labeling of behavior patterns could be inconsistent and error prone. This is because a human tends to interpret behavior based on the a priori cognitive knowledge of what should be present in a scene rather than solely based on what is visually detectable in the scene. This introduces a bias due to differences in experience and mental states.

It is worth pointing out that the proposed framework is by no means a general one, which can be applied to any type of scenarios. In particular, the proposed approach, as demonstrated by the experiments presented in Section 6, is able to cope with a moderately crowded scenario thanks to the discrete event-based behavior representation. However, an extremely busy and unstructured scenario such as an underground platform in rush hours will pose serious problems to the approach. This will be discussed in depth later in this paper.

The rest of the paper is structured as follows: Section 2 reviews related work to highlight the contributions of this work. Section 3 addresses the problem of behavior representation. The behavior profiling process is described in Section 4, which also explains how a composite generative behavior model can be built using a mixture of DBNs. Section 5 centers about the online detection of abnormal behavior and recognition of normal behavior using a behavior model. In Section 6, the effectiveness and robustness of our approach is demonstrated through experiments using noisy and sparse data sets collected from both indoor and outdoor surveillance scenarios. The paper concludes in Section 7.

2 Related Work

Much work on abnormal behavior detection[1] took a supervised learning approach [16], [14], [8], [5], [3] based on the assumption that there exist well-defined and known a priori behavior classes (both normal and abnormal). However, in reality, abnormal behavior is both rare and far from being well defined, resulting in insufficient clearly labeled data required for supervised model building.

More recently, a number of techniques have been proposed for unsupervised learning of behavior models [34], [9], [2], [28]. They can be further categorized into two different types according to whether an explicit model is built. Approaches that do not model behavior explicitly either perform clustering on observed patterns and label those forming small clusters as being abnormal [34], [9] or build a database of spatiotemporal patches using only regular/normal behavior and detect those patterns that cannot be composed from the database as being abnormal [2]. The approach proposed in [34] cannot be applied to any previously unseen behavior patterns and therefore is only suitable for postmortem analysis but not for on-the-fly anomaly detection. This problem is addressed by the approaches proposed in [9] and [2]. However, in these approaches, all the previously observed normal behavior patterns must be stored either in the form of histograms of discrete events [9] or ensembles of spatiotemporal patches [2] for detecting anomaly from unseen data, which jeopardizes the scalability of these approaches.

There is also another approach that differs from both the supervised and unsupervised techniques above. A semisupervised model was introduced in [33] with a two-stage training process. In stage one, a normal behavior model is learned using labeled normal patterns. In stage two, an abnormal behavior model is then learned unsupervised using

1. The notion of abnormal behavior appeared in different names in the literature including unusual, suspicious, or surprising behavior/events/activities, or simply anomaly, abnormality, irregularities, or outliers.

Bayesian adaptation. This approach still suffers from the laborious and inconsistent manual data labeling process.

In our work, an explicit model based on a mixture of DBNs is constructed in an unsupervised manner to learn specific behavior classes for automatic detection of abnormalities on the fly given unseen data. Compared to our previous work [28], we develop a more principled criterion for anomaly detection and normal behavior recognition based on a runtime accumulative anomaly measure and an online LRT method originally proposed for keywords detection in speech recognition [26]. This makes our approach more robust to noise in behavior representation. Our approach is similar to [9] in that behavior patterns are represented using discrete events. However, the manual labeling of objects of interests and manual event annotation are required in [9], which make it not fully unsupervised. Moreover, the behavior representation in [9] is based on an event histogram that ignores/throws away any information about the duration of an event. Such a behavior representation thus has less discriminative power compared to our method. Note that the anomaly detection method proposed in [9] was claimed to be online. Nevertheless, in [9], anomaly detection is performed only when the complete behavior pattern is observed, whereas in our work, it is performed on the fly. Our work is similar in spirit to [2] in that the behavior model (constructed in [2] as a database of video patches) is able to infer and generalize from the training data to unseen data. However, apart from the scalability problem mentioned above, the approach in [2] has limitations in capturing the temporal ordering aspect of a behavior pattern due to the constraint on the size of the video patches. In particular, the approach can only detect unusual local spatiotemporal formations from a single objects rather than subtle abnormalities embedded in the temporal correlations among multiple objects that are not necessarily close to each other in space and time. In summary, the key advantages of the proposed approach over previous approaches are as follows: 1) it is based on constructing a composite generative behavior model that scales well with the complexity of behavior and is robust to errors in behavior representation and 2) it performs on-the-fly anomaly detection and is therefore suitable for real-time surveillance applications.

3 Behavior Pattern Representation

3.1 Video Segmentation

The goal is to automatically segment a continuous video sequence $\mathbf{V}$ into N video segments $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_n, \ldots, \mathbf{v}_N\}$ such that, ideally, each segment contains a single behavior pattern. The nth video segment $\mathbf{v}_n$ consisting of T_n image frames is represented as $\mathbf{v}_n = [\mathbf{I}_{n1}, \ldots, \mathbf{I}_{nt}, \ldots, \mathbf{I}_{nT_n}]$, where $\mathbf{I}_{nt}$ is the tth image frame. Depending on the nature of the video sequence to be processed, various segmentation approaches can be adopted. Since we are focusing on surveillance video, the most commonly used shot change detection-based segmentation approach is not appropriate. In a not-too-busy scenario, there are often nonactivity gaps between two consecutive behavior patterns that can be utilized for activity segmentation. In the case where obvious nonactivity gaps are not available, the online segmentation algorithm proposed in [27] can be adopted. Specifically, video content is represented as a high-dimensional trajectory based on automatically detected visual events. Breakpoints on the trajectory are then detected online using a Forward-Backward Relevance (FBR) procedure. Alternatively, the video can be simply sliced into overlapping segments with a fixed time duration [34].

3.2 Event-Based Behavior Representation

First, an adaptive Gaussian mixture background model [24] is adopted to detect foreground pixels, which are modeled using Pixel Change History (PCH) [29]. Second, the foreground pixels in a vicinity are grouped into a blob using the connected component method. Each blob with an average PCH value greater than a threshold is then defined as a scene event. A detected scene event is represented as a seven-dimensional (7D) feature vector

$$\mathbf{f} = [\bar{x}, \bar{y}, w, h, R_f, M_p x, M_p y], \tag{1}$$

where $(\bar{x}, \bar{y})$ is the centroid of the blob, (w, h) is the blob dimension, R_f is the filling ratio of foreground pixels within the bounding box associated with the blob, and $(M_p x, M_p y)$ are a pair of first-order moments of the blob represented by PCH. Among these features, $(\bar{x}, \bar{y})$ are location features, (w, h) and R_f are principally shape features but also contain some indirect motion information, and $(M_p x, M_p y)$ are motion features capturing the direction of object motion.

Third, clustering is performed in the 7D scene event feature space using a Gaussian Mixture Model (GMM). The number of scene event classes K_e captured in the videos is determined by automatic model order selection based on the Bayesian Information Criterion (BIC) [21]. The learned GMM is used to classify each detected event into one of the K_e event classes. Finally, the behavior pattern captured in the nth video segment $\mathbf{v}_n$ is represented as a feature vector $\mathbf{P}_n$, given as

$$\mathbf{P}_n = [\mathbf{p}_{n1}, \ldots, \mathbf{p}_{nt}, \ldots, \mathbf{p}_{nT_n}], \tag{2}$$

where T_n is the length of the nth video segment, and the tth element of $\mathbf{P}_n$ is a K_e-dimensional variable

$$\mathbf{p}_{nt} = [p_{nt}^1, \ldots, p_{nt}^k, \ldots, p_{nt}^{K_e}]. \tag{3}$$

$\mathbf{P}_{nt}$ corresponds to the tth image frame of $\mathbf{v}_n$, where p_{nt}^k is the posterior probability that an event of the kth event class has occurred in the frame, given the learned GMM. If an event of the kth class is detected in the tth image frame of $\mathbf{v}_n$, we have $0 < p_{nt}^k \leq 1$; otherwise, we have $p_{nt}^k = 0$. Our event-based behavior representation is illustrated through an example in Fig. 1. Note that different classes of events occurred simultaneously (see Fig. 1a).

It is worth pointing out the following: 1) A behavior pattern is decomposed into temporally ordered semantically meaningful scene events. Instead of using low-level image features such as location, shape, and motion (1) directly for behavior representation, we represent a behavior pattern using the probabilities of different classes of events occurring in each frame. Consequently, the behavior representation is compact and concise. This is critical for a model-based behavior profiling approach because model construction based upon concise representation is more likely to be computationally tractable for complex behavior. 2) Different types of behavior patterns can differ either in the classes of events they are composed of or in the temporal orders of the event occurrence. For instance, behavior patterns A and B are deemed as being different if 1) A is composed of events of classes a, b, and d, whereas B is composed of events of classes a, c and e; or 2) both A and B are composed of events of classes

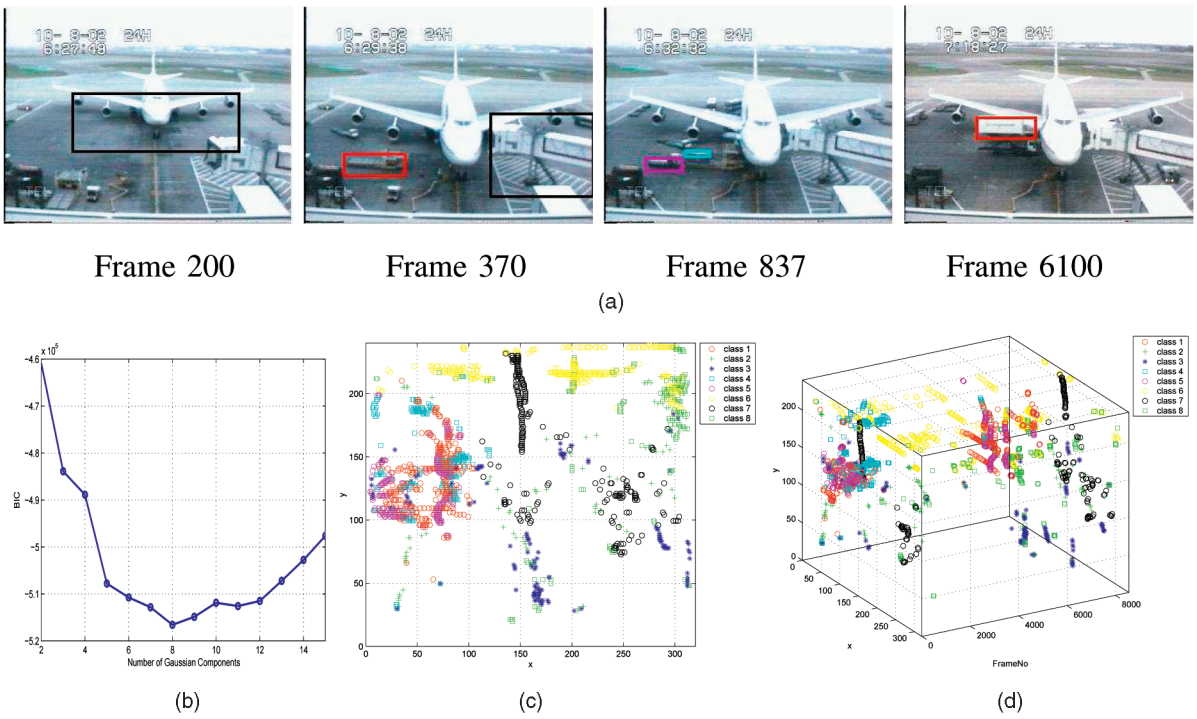


Fig. 1. Event-based behavior representation for an aircraft docking video. Details of the video can be found in Section 6.2. (b) shows that eight classes of events are detected automatically using BIC. Different classes of events are highlighted in the image frame using bounding boxes in different colors in (a). The spatial and temporal distribution of events of different classes throughout the sequence is illustrated in (c) and (d), respectively, with centroids of different classes of events depicted using the same color coding scheme as (a). In particular, events corresponding to the movements of objects involved in the front cargo and catering services are indicated in red, cyan, and magenta; events corresponding to a moving and stopping aircraft and airbridges are indicated in black and green (rectangular), respectively; and events corresponding to the movements of aircraft-pushing vehicles, passing-by vehicles in the back, and rear catering vehicles are in blue, yellow, and green (cross), respectively.

a, c and d; however, in A, event (class) a is followed by c, whereas in B, event (class) a is followed by d.

4 Behavior Profiling

The behavior profiling problem can now be defined formally. Consider a training data set $\mathbf{D}$ consisting of N feature vectors

$$\mathbf{D} = \{\mathbf{P}_1, \ldots, \mathbf{P}_n, \ldots, \mathbf{P}_N\}, \tag{4}$$

where $\mathbf{P}_n$ is defined in (2), representing the behavior pattern captured by the nth video segment $\mathbf{v}_n$. The problem to be addressed is to discover the natural grouping of the training behavior patterns upon which a model for normal behavior can be built. This is essentially a data clustering problem with the number of clusters unknown. There are a number of aspects that make this problem challenging: 1) Each feature vector $\mathbf{P}_n$ can be of different lengths. Conventional clustering approaches such as K-Means and mixture models require that each data sample is represented as a fixed length feature vector. These approaches thus cannot be applied directly. 2) A definition of a distance/affinity metric among these variable length feature vectors is nontrivial. Measuring affinity between feature vectors of variable length often involves Dynamic Time Warping (DTW) [11]. A standard DTW method used in computer vision community would attempt to treat the feature vector $\mathbf{P}_n$ as a K_e-dimensional trajectory and measure the distance of two behavior patterns by finding correspondence between discrete vertices on two trajectories. Since in our framework, a behavior pattern is represented as a set of temporal correlated events, that is, a stochastic process, a stochastic modeling-based approach is more appropriate for distance measuring. Note that in the case of matching two sequences of different lengths based on video object detection, the affinity of the most similar pair of images from two sequences can be used for sequence affinity measurement [22]. However, since we focus on the modeling behavior that could involve multiple objects interacting over space and time, the approach in [22] cannot be applied directly in our case. 3) Model selection needs to be performed to determine the number of clusters. To overcome the abovementioned difficulties, we propose a spectral clustering algorithm with feature and model selection based on modeling each behavior pattern using a DBN. Fig. 2 shows a diagrammatic illustration of our behavior profiling approach. It shows clearly that the proposed spectral clustering algorithm (blocks inside the dashed box) is the core of the approach. The key components of our approach are explained in details in the following sections.

4.1 Affinity Matrix

DBNs provide a solution for measuring the affinity between different behavior patterns. More specifically, each behavior pattern in the training set is modeled using a DBN. To measure the affinity between two behavior patterns represented as $\mathbf{P}_i$ and $\mathbf{P}_j$, two DBNs denoted as $\mathbf{B}_i$ and $\mathbf{B}_j$ are trained on $\mathbf{P}_i$ and $\mathbf{P}_j$, respectively, using the expectation-maximization (EM) algorithm [4], [7]. The affinity between $\mathbf{P}_i$ and $\mathbf{P}_j$ is then computed as

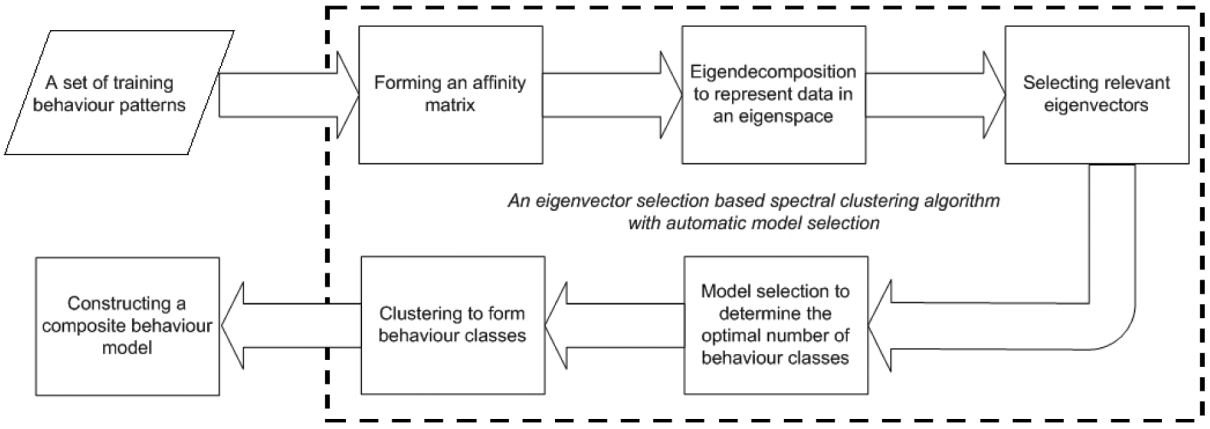


Fig. 2. A block diagram illustrating our behavior profiling approach.

$$S_{ij} = \frac{1}{2}\left\{\frac{1}{T_j}\log P(\mathbf{P}_j|\mathbf{B}_i) + \frac{1}{T_i}\log P(\mathbf{P}_i|\mathbf{B}_j)\right\}, \quad (5)$$

where $P(\mathbf{P}_j|\mathbf{B}_i)$ is the likelihood of observing $\mathbf{P}_j$ given $\mathbf{B}_i$, and T_i and T_j are the lengths of $\mathbf{P}_i$ and $\mathbf{P}_j$, respectively.[2]

DBNs of different topologies can be employed. A straightforward choice would be a Hidden Markov Model (HMM) (Fig. 3a). In this HMM, the observation variable at each time instance corresponds to $\mathbf{p}_{nt}$ (3), which represents the content of the tth frame of the nth behavior pattern, and is of dimension K_e, that is, the number of event classes. The conditional probability distributions (CPDs) of $\mathbf{p}_{nt}$ is assumed to be Gaussian for each of the N_s states of its parent node. However, a drawback of an HMM is that too many parameters are needed to describe the model when the observation variables are of high dimension. This makes an HMM vulnerable to overfitting, therefore generating poorly to unseen data. It is especially true in our case because an HMM needs to be learned for every single behavior pattern in the training data set, which could be short in duration. To solve this problem, we employ a Multiobservation HMM (MOHMM), [8] shown in Fig. 3b. Compared to an HMM, the observational space is factorized by assuming that each observed feature (p_{nt}^k) is independent of each other. Consequently, the number of parameters for describing a MOHMM is much lower than that for an HMM ($2K_eN_s + N_s^2 - 1$ for a MOHMM and $(K_e^2 + 3K_e)N_s/2 + N_s^2 - 1$ for an HMM). In this paper, N_s, the number of hidden states for each hidden variables in MOHMM, is set to K_e, that is, the number of event classes. This is reasonable because the value of N_s should reflect the complexity of a behavior pattern and so should the value of K_e.

An $N \times N$ affinity matrix $\mathbf{S} = [S_{ij}]$, where $1 \leq i$ and $j \leq N$, provides a new representation for the training data set, denoted as $\mathbf{D_s}$. In this representation, a behavior pattern is represented by its affinity to each behavior pattern in the training set. Specifically, the nth behavior pattern is now represented as the nth row of $\mathbf{S}$, denoted as $\mathbf{s}_n$. We thus have

$$\mathbf{D_s} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n, \ldots, \mathbf{s}_N\}. \quad (6)$$

Consequently, each behavior pattern is represented as a feature vector of fixed length N. Taking a conventional data clustering approach, model selection can be performed first to determine the number of clusters, which is then followed by data grouping using either a parametric approach such as Mixture Models or a nonparametric K-Nearest Neighbor model. However, since the number of data samples is equal to the dimensionality of the feature space, dimension reduction is necessary to avoid the "curse of dimensionality" problem. This is achieved through a novel spectral clustering algorithm that reduces the data dimensionality and performs clustering using the selected relevant eigenvectors of the data affinity matrix.

4.2 Eigendecomposition

Dimension reduction on the N-dimensional feature space defined in (6) can be achieved through the eigendecomposition of the affinity matrix $\mathbf{S}$. The eigenvectors of the affinity matrix are then used for data clustering. However, it has been shown in [25], [23] that it is more desirable to perform clustering using the eigenvectors of the normalized affinity matrix $\bar{\mathbf{S}}$, defined as

$$\bar{\mathbf{S}} = \mathbf{L}^{-\frac{1}{2}}\mathbf{S}\mathbf{L}^{-\frac{1}{2}}, \quad (7)$$

where $\mathbf{L} = [L_{ij}]$ is an $N \times N$ diagonal matrix with $L_{ii} = \sum_j S_{ij}$. It has been proven in [30], [25] that under certain constraints, the largest[3] K eigenvectors of $\bar{\mathbf{S}}$ (that is, eigenvectors with the largest eigenvalues) are sufficient to partition the data set into K clusters. Representing the data set using the K largest eigenvectors reduces the data dimensionality from N (that is, the number of behavior patterns) to K (that is, the number of behavior pattern classes). For a given K, standard clustering approaches such as K-Means or Mixture Models can be adopted. The remaining problem is to determine K, which is unknown. This is solved through automatic model selection.

4.3 Model Selection

We assume that the number of clusters K is between 1 and K_m. K_m is a number sufficiently larger than the true value of K. Suppose that we set $K_m = \frac{1}{5}N$, where N is the number of samples in the training data set. This is a reasonable assumption because as a rule of thumb, a more sparse data set (that is, $\frac{1}{5}N < K < K_m$) would make any data clustering algorithm unworkable. The training data set is now

2. Note that there are other ways to compute the affinity between two sequences modeled using DBNs [17], [18]. However, we found through our experiments that different affinity measures make little difference for our behavior profiling task.

3. The largest eigenvectors are eigenvectors whose corresponding eigenvalues are the largest in magnitude.

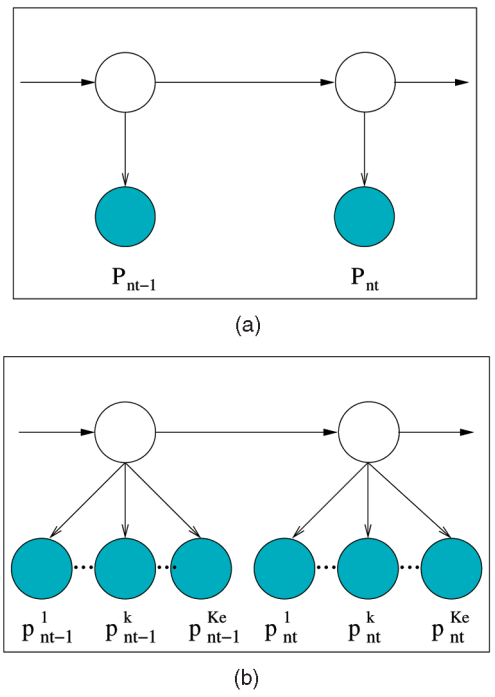


Fig. 3. Modeling a behavior pattern $\mathbf{P}_n = \{\mathbf{p}_{n1}, \ldots, \mathbf{p}_{nt}, \ldots, \mathbf{p}_{nT_n}\}$, where $\mathbf{p}_{nt} = \{p_{nt}^1, \ldots, p_{nt}^k, \ldots, p_{nt}^{K_e}\}$, using (a) an HMM and (b) a MOHMM. Observation nodes are shown as shaded circles and hidden nodes as clear circles.

represented using the K_m largest eigenvectors, denoted $\mathbf{D_e}$, as follows:

$$\mathbf{D_e} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n, \ldots, \mathbf{y}_N\} \tag{8}$$

with the nth behavior pattern being represented as a K_m-dimensional feature vector:

$$\mathbf{y}_n = [e_{1n}, \ldots, e_{kn}, \ldots, e_{K_m n}], \tag{9}$$

where e_{kn} is the nth element of the kth largest eigenvector $\mathbf{e_k}$. Since $K < K_m$, it is guaranteed that all the information needed for grouping K clusters is preserved in this K_m-dimensional feature space.

We model the distribution of $\mathbf{D_e}$ using a GMM. The log likelihood of observing the training data set $\mathbf{D_e}$ given a K-component GMM is computed as

$$\log P(\mathbf{D_e}|\theta) = \sum_{n=1}^{N}\left(\log \sum_{k=1}^{K} w_k P(\mathbf{y}_n|\theta_k)\right), \tag{10}$$

where $P(\mathbf{y}_n|\theta_k)$ defines the Gaussian distribution of the kth mixture component, and w_k is the mixing probability for the kth component. The model parameters θ are estimated using the EM algorithm. The BIC is then employed to select the optimal number of components K determining the number of behavior classes. For any given K, BIC is formulated as

$$BIC = -\log P(\mathbf{Y}|\theta) + \frac{C_K}{2}\log N, \tag{11}$$

where C_K is the number of parameters needed to describe a K-component Gaussian Mixture.

However, it is found in our experiments that in the $K_m = \frac{1}{5}N$-dimensional feature space, BIC tends to underestimate the number of clusters (see Fig. 4g for an example and more in Section 6). This is not surprising because BIC has been known for having the tendency of underfitting a model given sparse data [20]. A data set of N samples represented in a $K_m = \frac{1}{5}N$-dimensional feature space can always be considered as sparse for data clustering, as well as for determining the number of clusters. Our solution to this problem is to reduce the dimensionality through unsupervised feature selection, that is, selecting relevant/informative eigenvectors of the normalized affinity matrix to perform model selection and data clustering.

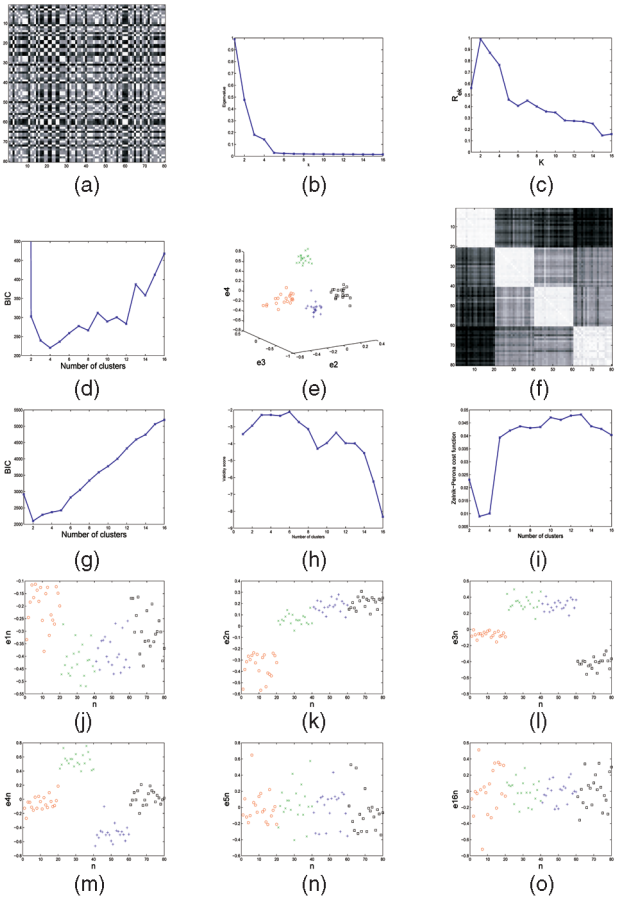


Fig. 4. Clustering a synthetic data set using our spectral clustering algorithm. The eigenvalues of the $K_m = 16$ largest eigenvectors is shown in (b). (c) depicts the learned relevance scores. The first four largest eigenvectors were determined as being relevant using (13). (d) shows the BIC model selection results; the optimal cluster number was determined as 4. (e) shows the 80 data samples plotted using the three most relevant eigenvectors, that is, $\mathbf{e}_2$, $\mathbf{e}_3$, and $\mathbf{e}_4$. Points corresponding to different classes are color coded in (e) according to the classification result. (f) shows the affinity matrix reordered according to the result of our clustering algorithm. (g), (h), and (i) show that the cluster number was estimated as 2, 5, and 3, respectively, using three alternative approaches. The distribution of some of the top 16 eigenvectors is shown in (j), (k), (l), (m), (n), and (o). (a) Normalized affinity matrix. (b) Eigenvalues corresponding to top eigenvectors. (c) Learned eigenvector relevance. (d) BIC with eigenvector selection. (e) Data distribution in $\mathbf{e}_2$, $\mathbf{e}_3$, and $\mathbf{e}_4$. (f) Affinity matrix reordered after clustering. (g) BIC without eigenvector selection. (h) Validity score. (i) Zelnik-Perona cost function. (j) $\mathbf{e}_1$. (k) $\mathbf{e}_2$. (l) $\mathbf{e}_3$. (m) $\mathbf{e}_4$. (n) $\mathbf{e}_5$. (o) $\mathbf{e}_{16}$.

4.4 Eigenvector Selection for Behavior Pattern Clustering

We aim to derive a suitable criterion for measuring the relevance of each eigenvector of the normalized affinity matrix $\bar{\mathbf{S}}$. Intrinsically, only a subset of the K_m largest

eigenvectors are relevant for grouping K clusters. An eigenvector is deemed as being relevant for clustering if it can be used to separate at least one cluster of data from the others. It is necessary and crucial to identify and remove those irrelevant/uninformative eigenvectors not only because we need to reduce the dimensionality of the feature space but also due to the factor that irrelevant features degrade the accuracy of learning and, therefore, the performance of clustering.

An intuitive solution to measuring the eigenvector relevance would be to investigate the associated eigenvalue for each eigenvector. The analysis in [15] shows that in an "ideal" case where different clusters are infinitely far apart, the top K_{true} (relevant) eigenvectors have a corresponding eigenvalue of magnitude 1 and others do not. In this case, simply selecting those eigenvectors would solve the problem. In fact, the estimation of the number of clusters also becomes trivial by looking at the eigenvalues: it is equal to the number of eigenvalues of magnitude 1. Indeed, eigenvalues are useful when the data are clearly separated, that is, close to the "ideal" case. However, given a more realistic dataset the eigenvalues are not useful as relevant eigenvectors do not necessarily assume high eigenvalues and higher eigenvalues do not necessarily mean higher relevance either (see Fig. 4). Next, we derive a novel eigenvector relevance learning algorithm based on measuring the relevance of an eigenvector according to how well it can separate a data set into different clusters. This is achieved through modeling the distributions of the elements of each eigenvector with considerations on the following a priori knowledge about the affinity matrix eigenspace: 1) The distribution of the elements of a relevant eigenvector must enable it to be used for separating at least one cluster from the others. More specifically, the distribution of its elements is multimodal if it is relevant and unimodal otherwise. 2) A large eigenvector is more likely to be relevant in data clustering than a small one.

We denote the likelihood of the kth largest eigenvector $\mathbf{e_k}$ being relevant as $R_{\mathbf{e_k}}$, where $0 \leq R_{\mathbf{e_k}} \leq 1$. We assume that the elements of $\mathbf{e_k}$, e_{kn} follow two different distributions, namely, unimodal and multimodal, depending on whether $\mathbf{e_k}$ is relevant. The probability density function (pdf) of e_{kn} is thus formulated as a mixture model of two components

$$p(e_{kn}|\theta_{e_{kn}}) = (1 - R_{\mathbf{e_k}})p\Big(e_{kn}|\theta^1_{e_{kn}}\Big) + R_{\mathbf{e}_k}p\Big(e_{kn}|\theta^2_{e_{kn}}\Big),$$

where $\theta_{e_{kn}}$ are the parameters describing the distribution, and $p(e_{kn}|\theta^1_{e_{kn}})$ is the pdf of e_{kn} when $\mathbf{e_k}$ is irrelevant/uninformative and $p(e_{kn}|\theta^2_{e_{kn}})$ otherwise. $R_{\mathbf{e_k}}$ acts as the weight or mixing probability of the second mixture component. The distribution of e_{kn} is assumed to be a single Gaussian (unimodal) to reflect the fact that $\mathbf{e_k}$ cannot be used for data clustering when it is irrelevant

$$p(e_{kn}|\theta^1_{e_{kn}}) = \mathcal{N}(e_{kn}|\mu_{k1}, \sigma_{k1}),$$

where $\mathcal{N}(.|\mu, \sigma)$ denotes a Gaussian of mean μ and covariance σ. We assume the second component of $P(\mathbf{e_k}|\theta_{\mathbf{e_k}})$ as a mixture of two Gaussians to reflect the fact that $\mathbf{e_k}$ can separate one cluster of data from the others when it is relevant

$$p(e_{kn}|\theta^2_{e_{kn}}) = w_k\mathcal{N}(e_{kn}|\mu_{k2}, \sigma_{k2}) + (1 - w_k)\mathcal{N}(e_{kn}|\mu_{k3}, \sigma_{k3}),$$

where w_k is the weight of the first Gaussian in $p(e_{kn}|\theta^2_{e_{kn}})$. There are two reasons for using a mixture of two Gaussians even when e_{kn} forms more than two clusters or the distribution of each cluster is not Gaussian: 1) in these cases, a mixture of two Gaussians ($p(e_{kn}|\theta^2_{e_{kn}})$) still fits better to the data compared to a single Gaussian ($p(e_{kn}|\theta^1_{e_{kn}})$) and 2) its simple form means that only a small number of parameters are needed to describe $p(e_{kn}|\theta^2_{e_{kn}})$. This makes model learning possible even when given sparse data.

There are eight parameters required for describing the distribution of e_{kn}

$$\theta_{e_{kn}} = \{R_{\mathbf{e_k}}, \mu_{k1}, \mu_{k2}, \mu_{k3}, \sigma_{k1}, \sigma_{k2}, \sigma_{k3}, w_k\}. \quad (12)$$

The ML estimate of $\theta_{e_{kn}}$ can be obtained using the following algorithm. First, the parameters of the first mixture component $\theta^1_{e_{kn}}$ are estimated as $\mu_{k1} = \frac{1}{N}\sum_{n=1}^{N} e_{kn}$ and $\sigma_{k1} = \frac{1}{N}\sum_{n=1}^{N}(e_{kn} - \mu_{k1})^2$. The remaining six parameters are then estimated iteratively using EM [4]. It is important to note that $\theta_{e_{kn}}$ as a whole are *not* estimated iteratively using a standard EM algorithm, although EM was employed for part of $\theta_{e_{kn}}$, namely, $\theta^1_{e_{kn}}$. This is critical for our algorithm because if all the eight parameters are re-estimated in each iteration, the distribution of e_{kn} is essentially modeled as a mixture of three Gaussians, and the estimated $R_{\mathbf{e_k}}$ would represent the weight of two of the three Gaussians. This is very different from what $R_{\mathbf{e_k}}$ is meant to represent, that is, the likelihood of $\mathbf{e_k}$ being relevant for data clustering.

Since our relevance learning algorithm is essentially a local (greedy) searching method, the algorithm could be sensitive to parameter initialization, especially when given noisy and sparse data [4]. To overcome this problem, first, our a priori knowledge on the relationship between the relevance of each eigenvector and its corresponding eigenvalue is utilized to set the initial value of $R_{\mathbf{e_k}}$. Specifically, we set $\tilde{R_{\mathbf{e_k}}} = \bar{\lambda}_k$, where $\tilde{R_{\mathbf{e_k}}}$ is the initial value of $R_{\mathbf{e_k}}$, and $\bar{\lambda}_k \in [0, 1]$ is the normalized eigenvalue for $\mathbf{e_k}$ with $\bar{\lambda}_1 = 1$ and $\bar{\lambda}_{K_m} = 0$. We then randomly initialize the values of the other five parameters, namely, μ_{k2}, μ_{k3}, σ_{k2}, σ_{k3}, and w_k, and the solution that yields the largest $p(e_{kn}|\theta^2_{e_{kn}})$ over different initializations is chosen.

It is worth pointing out that although our relevance learning algorithm is based on estimating the distribution of the elements of each eigenvector, we are only interested in learning whether the distribution is unimodal or multimodal, which is reflected by the value of $R_{\mathbf{e_k}}$. In other words, among the eight free parameters of the eigenvector distribution (12), $R_{\mathbf{e_k}}$ is the only parameter that we are after. This also explains why our algorithm is able to estimate the relevance accurately when there are more than two clusters and/or the distribution of each cluster is not Gaussian.

The ML estimate $\hat{R_{\mathbf{e_k}}}$ thus provides a real-value measurement of the relevance of $\mathbf{e_k}$. Since a "hard decision" is needed for dimension reduction, we eliminate the kth eigenvector $\mathbf{e_k}$ among the K_m candidate eigenvectors if

$$\hat{R_{\mathbf{e_k}}} < 0.5. \quad (13)$$

After eliminating those irrelevant eigenvectors, the selected relevant eigenvectors are used to determine the number of clusters K and perform clustering based on GMM and BIC as described in Section 4.3. Each behavior pattern in the training data set is then labeled as one of the K behavior classes.

Fig. 4 shows an example of data clustering using our eigenvector selection-based spectral clustering algorithm. A total of 80 sequences were randomly generated using four MOHMMs. The data set is composed of 20 sequences sampled from each MOHMM. The lengths of these segments

were set randomly, ranging from 200 to 600. The four MOHMMs have the same topology as the one shown in Fig. 3b with different parameters set randomly. It can be seen in Figs. 4j, 4k, 4l, 4m, 4n, and 4o that the second, third, and fourth eigenvectors contain strong information about the grouping of data, whereas the largest eigenvector is much less informative. The remaining eigenvectors contain virtually no information (see Figs. 4n and 4o). It can be seen in Fig. 4c that the proposed relevance measure $R_{\mathbf{e_k}}$ accurately reflects the relevance of each eigenvectors. By thresholding the relevance measure (13), the four largest eigenvectors are kept for clustering. Fig. 4e shows that the four clusters are clearly separable in the eigenspace spanning the three most relevant eigenvectors. It is thus not surprising that the number of clusters was determined correctly as four using BIC on the relevant eigenvectors (see Fig. 4d). The clustering result is illustrated using the reordered affinity matrix in Fig. 4f, showing that all four clusters were discovered accurately. We also estimated the number of clusters using three alternative methods: 1) BIC using all 16 eigenvectors, 2) Porikli and Haga's validity score [19] (maximum score corresponding to the optimal number), and 3) Zelnik-Perona cost function [32] (minimum cost corresponding to the optimal number). Figs. 4g, 4h, and 4i show that none of these methods was able to yield an accurate estimate of the cluster number.

4.5 A Composite Behavior Model Using a Mixture of MOHMMs

To build a model for the observed/expected behavior, we first model the kth behavior class using a MOHMM $\mathbf{B}_k$. The parameters of $\mathbf{B}_k$, $\theta_{\mathbf{B}_k}$ are estimated using all the patterns in the training set that belong to the kth class. A behavior model $\mathbf{M}$ is then formulated as a mixture of the K MOHMMs. Given an unseen behavior pattern, represented as a behavior pattern feature vector $\mathbf{P}$ as described in Section 3, the likelihood of observing $\mathbf{P}$ given $\mathbf{M}$ is

$$P(\mathbf{P}|\mathbf{M}) = \sum_{k=1}^{K} \frac{N_k}{N} P(\mathbf{P}|\mathbf{B}_k), \tag{14}$$

where N is the total number of training behavior patterns and N_k is the number of patterns that belong to the kth behavior class.

5 Online Anomaly Detection and Normal Behavior Recognition

Once constructed, the composite behavior model $\mathbf{M}$ can be used to detect whether an unseen behavior pattern is normal using a runtime anomaly measure. If it is detected to be normal, the behavior pattern is then recognized as one of the K classes of normal behavior patterns using an online LRT method.

An unseen behavior pattern of length T is represented as $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_t, \ldots, \mathbf{p}_T]$. At the tth frame, the accumulated visual information for the behavior pattern, represented as $\mathbf{P}_t = [\mathbf{p}_1, \ldots, \mathbf{p}_t]$, is used for online reliable anomaly detection. First, the normalized log likelihood of observing $\mathbf{P}$ at the tth frame given the behavior model $\mathbf{M}$ is computed as

$$l_t = \frac{1}{t} \log P(\mathbf{P}_t|\mathbf{M}). \tag{15}$$

l_t can be easily computed online using the forward-backward procedure [13]. Specifically, to compute l_t, the K_e forward probabilities at time t are computed using the K_e forward probabilities computed at time $t-1$, together with the observations at time t (see [13] for details). Note that the complexity of computing l_t is $\mathcal{O}(K_e^2)$ and does not increase with t.

We then measure the anomaly of $\mathbf{P}_t$ using an online anomaly measure Q_t

$$Q_t = \begin{cases} l_1 & \text{if } t = 1 \\ (1-\alpha)Q_{t-1} + \alpha(l_t - l_{t-1}) & \text{otherwise,} \end{cases} \tag{16}$$

where α is an accumulating factor determining how important the visual information extracted from the current frame is for anomaly detection. We have $0 < \alpha \leq 1$. Compared to l_t as an indicator of normality/anomaly, Q_t could add more weight to more recent observations. Anomaly is detected at frame t if

$$Q_t < Th_A, \tag{17}$$

where Th_A is the anomaly detection threshold. The value of Th_A should be set according to the detection and false alarm rates required by each particular surveillance application. Note that it takes a time delay for Q_t to stabilize at the beginning of evaluating a behavior pattern due to the nature of the forward-backward procedure. The length of this time period, denoted as T_w, is related to the complexity of the MOHMM used for behavior modeling. We thus set $T_w = 3K_e$ in our experiments to be reported later in Section 6, that is, the anomaly of a behavior pattern is only evaluated when $t > T_w$.

At each frame t, a behavior pattern needs to be recognized as one of the K behavior classes when it is detected as being normal, that is, $Q_t > Th_A$. This is achieved by using an online LRT method. More specifically, we consider a hypotheses test between the following

H_k: $\mathbf{P}_t$ is from the hypothesized model $\mathbf{B}_k$ and belongs to the kth normal behavior class;

H_0: $\mathbf{P}_t$ is from a model other than $\mathbf{B}_k$ and does not belong to the kth normal behavior class;

where H_0 is called the alternative hypothesis. Using LRT, we compute the likelihood ratio of accepting the two hypotheses as

$$r_k = \frac{P(\mathbf{P}_t; H_k)}{P(\mathbf{P}_t; H_0)}. \tag{18}$$

The hypothesis H_k can be represented by the model $\mathbf{B}_k$, which has been learned in the behavior profiling step. The key to LRT is thus to construct the alternative model that represents H_0. In a general case, the number of possible alternatives is unlimited; $P(\mathbf{P}_t; H_0)$ can thus only be computed through approximation [26], [10]. Fortunately, in our case, we have determined at the tth frame that $\mathbf{P}_t$ is normal and can only be generated by one of the K normal behavior classes. Therefore, it is reasonable to construct the alternative model as a mixture of the remaining $K-1$ normal behavior classes. In particular, (18) is rewritten as

$$r_k = \frac{P(\mathbf{P}_t|\mathbf{B}_k)}{\sum_{i \neq k} \frac{N_i}{N - N_k} P(\mathbf{P}_t|\mathbf{B}_i)}. \tag{19}$$

Note that r_k is a function of t and computed over time. $\mathbf{P}_t$ is reliably recognized as the kth behavior class only when $1 \ll Th_r < r_k$. In our experiments, we found that $Th_r = 10$ led to satisfactory results. When there are more than one r_k

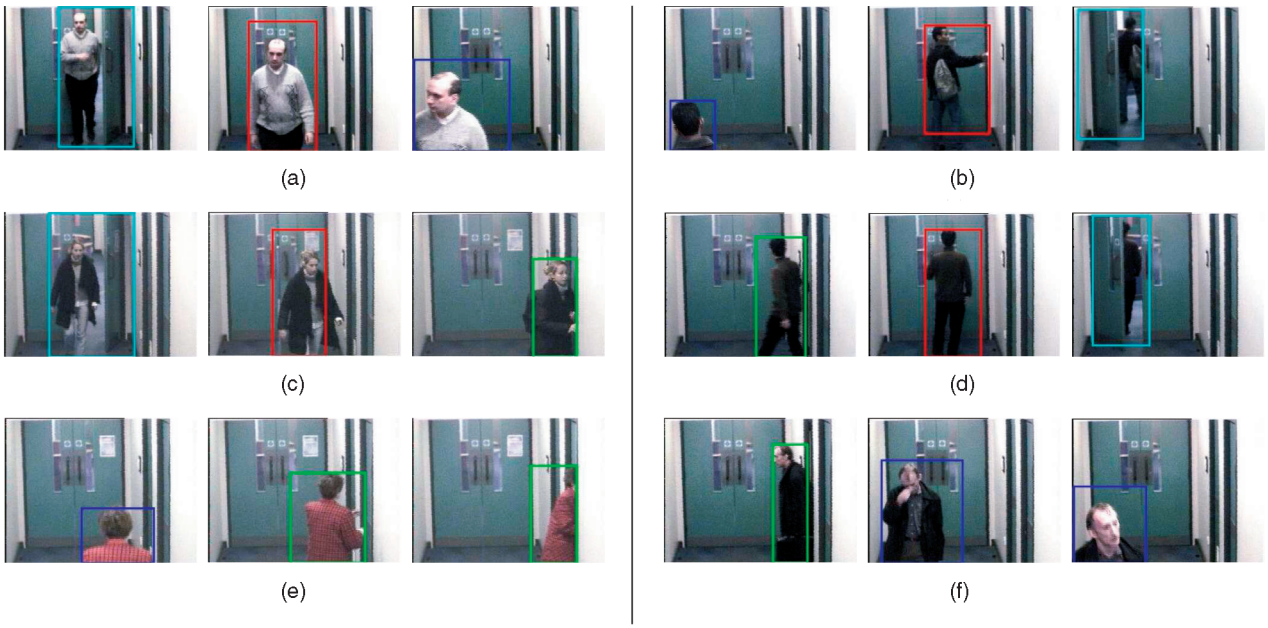


Fig. 5. Behavior patterns in a corridor entrance/exit scene. (a), (b), (c), (d), (e), and (f) show image frames of typical behavior patterns belonging to the six behavior classes listed in Table 1. The four classes of events detected automatically, "entering/leaving the near end of the corridor," "entering/leaving the entry door," "entering/leaving the side doors," and "in corridor with the entry door closed," are highlighted in the image frames using bounding boxes in blue, cyan, green, and red, respectively. The same color scheme will be used for illustrating detected events in Figs. 8 and 9. (a) C1. (b) C2. (c) C3. (d) C4. (e) C5. (f) C6.

greater than Th_r, the behavior pattern is recognized as the class with the largest r_k.

For comparison, the commonly used ML method recognizes $\mathbf{P}_t$ as the kth behavior class when $k = \arg\max_k \{P(\mathbf{P}_t|\mathbf{B}_k)\}$. Using the ML method, recognition has to be performed at each single frame without considering how reliable and sufficient the accumulated visual evidence is. This often causes errors especially when there are ambiguities between different classes (for example, a behavior pattern can be explained away equally well by multiple plausible behavior at its early stage). Compared to the ML method, our online LRT method holds the decision on behavior recognition until sufficient evidence has been accumulated to overcome ambiguities. The recognition results obtained using our approach are thus more reliable compared to those obtained by the ML method. Another commonly used method for classification is the Maximum A Posteriori (MAP) method. Although bearing a resemblance to our online LRT in formulation, the classification using the MAP rule is conceptually very different from that using a hypothesis test in LRT. In particular, unlike MAP, which has a standard formulation, LRT can be formulated differently, depending on how the likelihood of accepting the alternative hypothesis is computed (that is, the denominator of (18)). The main difference between LRT and MAP again lies in the fact that in a real-time application, LRT works better when multiple candidate behavior classes give equally plausible explanations for a temporally incomplete behavior pattern. In this case, the likelihood ratio value will be low and no decision will be made using LRT. In contrast with LRT, without having any a priori knowledge about the occurrence of different candidate behavior classes (that is, $P(\mathbf{B}_k) = 1/K$ with $1 \leq k \leq K$), the MAP values for multiple classes can be high, which will lead to a premature and wrong decision. Note that it is possible to modify the standard ML and MAP rules so that a recognition decision is withhold until more information is accumulated. Nevertheless, the difference mentioned above makes LRT advantageous for our behavior recognition task.

6 Experiments

In this section, we illustrate the effectiveness and robustness of our approach on behavior profiling and online anomaly detection with experiments using data sets collected from both indoor and outdoor surveillance scenarios.

6.1 Corridor Entrance/Exit Human Behavior Monitoring

Data set and feature extraction. A CCTV camera was mounted on the ceiling of an office entrance/exit corridor, monitoring the people entering and leaving an office area (see Fig. 5). The office area is secured by an entrance door that can only be opened by scanning an entry card on the wall next to the door (see middle frame in Fig. 5b). Two side doors were also located at the right-hand side of the corridor. People from both inside and outside the office area have access to those two side doors. Typical behavior occurring in the scene would be people entering or leaving either the office area or the side doors and walking toward the camera. Each behavior pattern would normally last a few seconds. For this experiment, a data set was collected over five different days consisting of 6 hours of video, totaling to 432,000 frames captured at 20 Hz with 320×240 pixels per frame. This data set was then segmented into sections separated by any motionless intervals lasting for more than 30 frames. This resulted in 142 video segments of actual behavior pattern instances. Each segment has on the average 121 frames, with the shortest having 42 frames and longest having 394 frames.

Model training. A training set consisting of 80 video segments was randomly selected from the overall 142 segments without any behavior class labeling of the video

TABLE 1
The Six Classes of Behavior Patterns that Most Commonly Occurred in the Corridor Entrance/Exit Scene

C1	From the office area to the near end of the corridor
C2	From the near end of the corridor to the office area
C3	From the office area to the side-doors
C4	From the side-doors to the office area
C5	From the near end of the corridor to the side-doors
C6	From the side-doors to the near end of the corridor

segments. The remaining 62 segments were used for testing the trained model later. This model training exercise was repeated 20 times, and in each trial, a different model was trained using a different random training set. This is in order to avoid any bias in the anomaly detection and normal behavior recognition results to be discussed later. For comparative evaluation, alternative models were also trained using labeled data sets as follows. For each of the 20 training sessions above, a model was trained using identical training sets as above. However, each behavior pattern in the training sets was also manually labeled as one of the manually identified behavior classes. On the average, 12 behavior classes were manually identified for the labeled training sets in each trial. Six classes were always identified in each training set (see Table 1). On the average, they accounted for 83 percent of the labeled training data.

Event detection for behavior representation. Given a training set, discrete events were detected and classified using automatic model order selection in clustering, resulting in four classes of events corresponding to the common constituents of all behavior in this scene: "entering/leaving the near end of the corridor," "entering/leaving the entrance door," "entering/leaving the side doors," and "in corridor with the entrance door closed." Examples of detected events are shown in Fig. 5 using color-coded bounding boxes. Note that it may appear to be intuitive and perhaps also desirable to have a "swipe card" event class for anomaly detection in this scenario. However, it is also observed that due to the view angle, most instances of the possible "swipe card" event are not visible in the scene (for example, occluded by human body). Moreover, different people could have very different ways of swiping a card particularly as a card is not required to make physical contact with the swipe machine in order to trigger a reading. Therefore, even if a supervised event detection approach is taken to artificially impose such an event class, the "swipe card" event could not be detected reliably. Using our unsupervised method, a likely "swipe-card" event is in effect classified into a general event class "in corridor with the entrance door closed" (see Fig. 5b for an example). Nevertheless, the experiments presented below demonstrate that these four general event classes enable our approach to detect anomaly robustly mainly by examining the temporal correlations of different events. It is also observed that that due to the narrow view nature of the scene, differences between the four common events are rather subtle and can be misidentified based on local information (space and time) alone, resulting in a large error margin in event detection. The fact that these events are also common constituents to different behavior patterns reinforces the assumption that local events treated in isolation hold little discriminative information for behavior profiling.

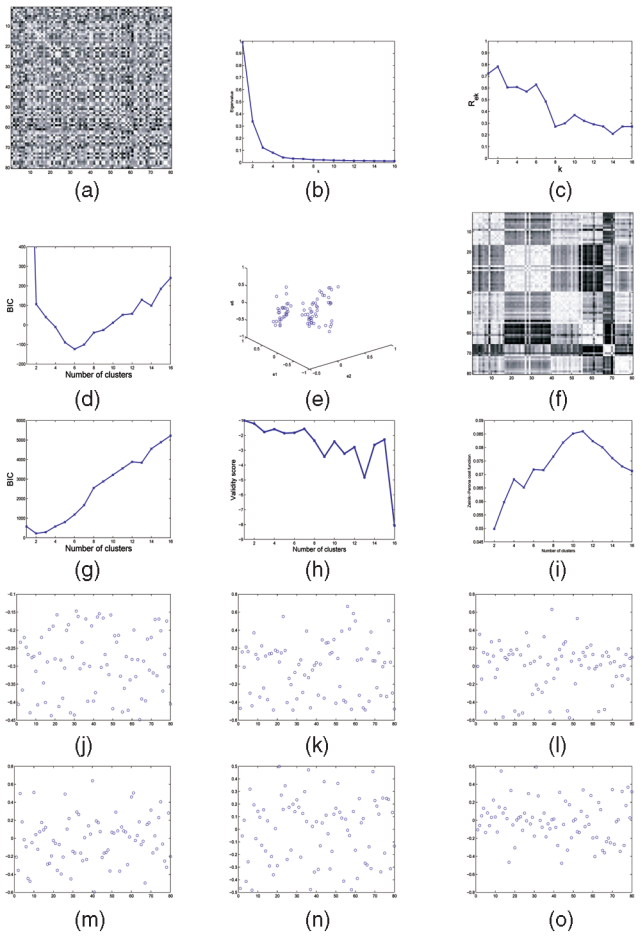


Fig. 6. An example of model training. (c) shows that the top six largest eigenvectors were determined as relevant features for clustering. (d) and (g) show that the number of behavior classes was determined as 6 and 2 using BIC with and without relevant eigenvector selection, respectively. (h) and (i) show that using Porikli and Haga's validity score and Zelnik-Manor and Perona's cost function, the class number was estimated as 1 and 2, respectively. (a) Normalized affinity matrix. (b) Eigenvalues corresponding to top eigenvectors. (c) Learned eigenvector relevance. (d) BIC with eigenvector selection. (e) Data distribution in $\mathbf{e}_2$, $\mathbf{e}_3$, and $\mathbf{e}_4$. (f) Affinity matrix reordered after clustering. (g) BIC without eigenvector selection. (h) Validity score. (i) Zelnik-Perona cost function. (j) $\mathbf{e}_1$. (k) $\mathbf{e}_2$. (l) $\mathbf{e}_3$. (m) $\mathbf{e}_4$. (n) $\mathbf{e}_6$. (o) $\mathbf{e}_{16}$.

Model training using unlabeled data. Over the 20 trials, on the average, six eigenvectors were automatically determined as being relevant for clustering, with the smallest being four and the largest being nine. It is noted that all the selected eigenvectors were among the 10 largest eigenvectors of the normalized affinity matrices. The number of clusters for each training set was determined automatically as six in every trial. By observation, each discovered data cluster mainly contained samples corresponding to one of the six behavior classes listed in Table 1. In comparison, all three alternative approaches, including BIC without eigenvector selection, Porikli and Haga's validity score, and Zelnik-Manor and Perona's cost function, tended to severely underestimate the class number. Fig. 6 shows an example of behavior pattern clustering using unlabeled training sets. Note that compared to the synthetic data (see Fig. 4), the data we have for behavior profiling is much more noisy and difficult to group. This is reflected by the fact that the elements of the eigenvectors show less information about the data grouping (see Figs. 6j, 6k, 6l, 6m, 6n, and 6o). However, using only the

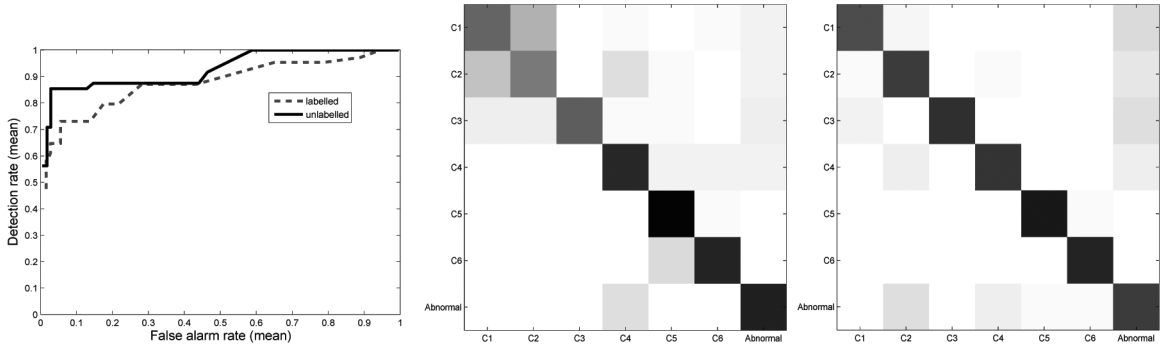


Fig. 7. (a) Comparing the performance of anomaly detection using corridor entrance/exit behavior models trained by labeled and unlabeled data. The mean ROC curves were obtained over 20 trials. (b) and (c) Comparing the performance of behavior recognition using models trained by labeled and unlabeled data. The two confusion matrices were obtained by averaging the results over 20 trials with $Th_A = -0.2$. Each row represents the probabilities of the corresponding class being confused with all the other classes, averaged over 20 trials.

relevant/informative eigenvectors, our algorithm can still discover the behavior classes correctly. For each unlabeled training set, a normal behavior model was constructed as a mixture of six MOHMMs, as described in Section 4.5.

Model training using labeled data. For each labeled training set, a normal behavior model was built as a mixture of MOHMMs with the number of mixture components determined by the number of manually identified behavior classes. Each MOHMM component was trained using the data samples corresponding to one class of manually identified behavior in each training set.

Anomaly detection. The behavior models built using both labeled and unlabeled behavior patterns were used to perform online anomaly detection. To measure the performance of the learned models on anomaly detection, each behavior pattern in the testing sets was manually labeled as normal if there were similar patterns in the corresponding training sets and abnormal otherwise. A testing pattern was detected as being abnormal when (17) was satisfied at any time after $T_w = 3K_e = 12$ frames. The accumulating factor α for computing Q_t was set to 0.1. We measure the performance of anomaly detection using the anomaly detection rate and the false alarm rate,[4] which are defined as

$$\begin{aligned}&\text{Anomaly-detection rate} = \\ &\frac{\#\text{ True positives (abnormal detected as abnormal)}}{\#\text{ All positives (abnormal patterns) in a data set}}, \\ &\text{False-alarm rate} = \\ &\frac{\#\text{ False positives (normal detected as abnormal)}}{\#\text{ All negatives (normal patterns) in a data set}}.\end{aligned} \tag{20}$$

The detection rate and false alarm rate of anomaly detection are shown in the form of a Receiver Operating Characteristic (ROC) curve by varying the anomaly detection threshold Th_A (see (17)). Fig. 7a shows that the models trained using unlabeled data clearly outperformed those trained using labeled data. In particular, it is found that given the same Th_A, the models trained using unlabeled data sets achieved a higher anomaly detection rate and a lower false alarm rate compared to those trained using labeled data sets (see also Table 2 and the last columns of the confusion matrices shown in Figs. 7b and 7c). Fig. 8 shows examples of false alarm and

TABLE 2
The Mean and Standard Deviation of the Anomaly Detection Rate and False Alarm Rates for Corridor Entrance/Exit Behavior Models Trained Using Unlabeled and Labeled Data

	Anomaly Detection Rate (%)	False Alarm Rate (%)
Unlabelled	85.4 ± 2.9	6.1 ± 3.1
Labelled	73.1 ± 12.9	8.4 ± 5.3

The results were obtained over 20 trials with $Th_A = -0.2$.

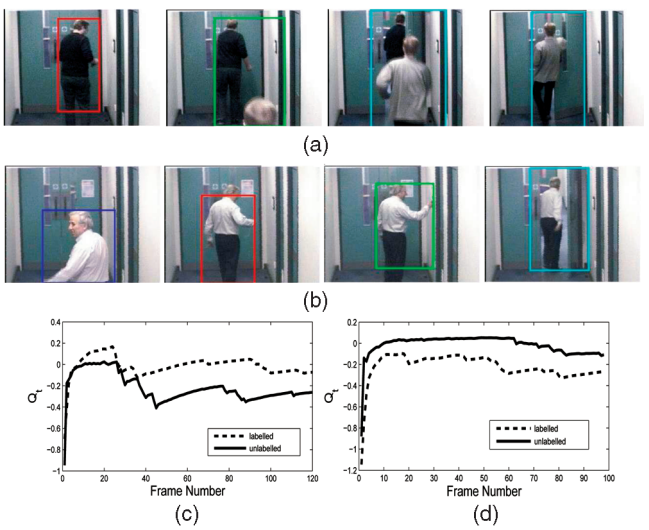


Fig. 8. Examples of anomaly detection in the corridor entrance/exit scene. (a) and (c) An abnormal behavior pattern was detected as being abnormal by the model trained using an unlabeled data set, whereas it was detected as being normal by the model trained using the same but labeled data set. It shows a person sneaking into the office area without using an entry card. (b) and (d) A normal behavior pattern that was detected correctly by the model trained using an unlabeled data set but detected as being abnormal by the model trained using the same but labeled data set. The third frame from left in (b) shows an error in event detection (an "in corridor with the entrance door closed" event was detected as an "entering/leaving the side doors" event). Note that a smaller value of Q_t means that it is more likely for the behavior pattern to be abnormal. Th_A was set to -0.2.

4. The anomaly detection rate and false alarm rate are also called true positive rate and false positive rate, respectively, in the literature. Note that the performance can also be measured using the true negative rate and the false negative rate. Since we have "true negative rate" = 1-"false alarm rate" and "false negative rate" = 1-"anomaly detection rate", showing only the anomaly detection rate and the false alarm rate is adequate.

TABLE 3
Comparing the Performance of Models Trained Using Unlabeled and Labeled Data on Corridor Normal Behavior Recognition

	Normal Behaviour Recognition Rate (%)
Unlabelled	77.9 ± 4.8
Labelled	84.7 ± 6.0

The results were obtained with $Th_A = -0.2$.

misdetection by models trained using labeled data. It is noted that the lower tolerance toward event detection errors was the main reason for the higher false alarm rate of models trained using labeled data (see Figs. 8b and 8d for an example).

Recognition of normal behavior. To measure the recognition rate, the normal behavior patterns in the testing sets were manually labeled into different behavior classes. A normal behavior pattern was recognized correctly if it was detected as normal and classified into a behavior class containing similar behavior patterns in the corresponding training set by the learned behavior model. We first compare the performance of models trained using labeled and unlabeled data. Table 3 shows that the models trained using labeled data achieved slightly higher recognition rates compared to those trained using unlabeled data. To have a more complete picture of the performance on normal behavior recognition, the standard confusion matrix is utilized. Fig. 7b shows that when a normal behavior pattern was not recognized correctly by a model trained using unlabeled data, it was most likely to be recognized as belonging to another normal behavior class. On the other hand, Fig. 7c shows that for a model trained by labeled data, a normal behavior pattern was most likely to be wrongly detected as an anomaly if it was not recognized correctly. This contributed to the higher false alarm rate for the model trained by labeled data.

Our online LRT method was also compared with the conventional ML method for online normal behavior recognition using unlabeled data trained models. Examples are shown in Fig. 9. It is noted that based on our online LRT method, normal behavior patterns were reliably and promptly recognized after sufficient visual evidence had become available (see Figs. 9c and 9g). On the contrary, based on the ML method, decisions on behavior recognition were made prematurely and unreliably due to the ambiguities among different behavior classes (see Figs. 9d and 9h).

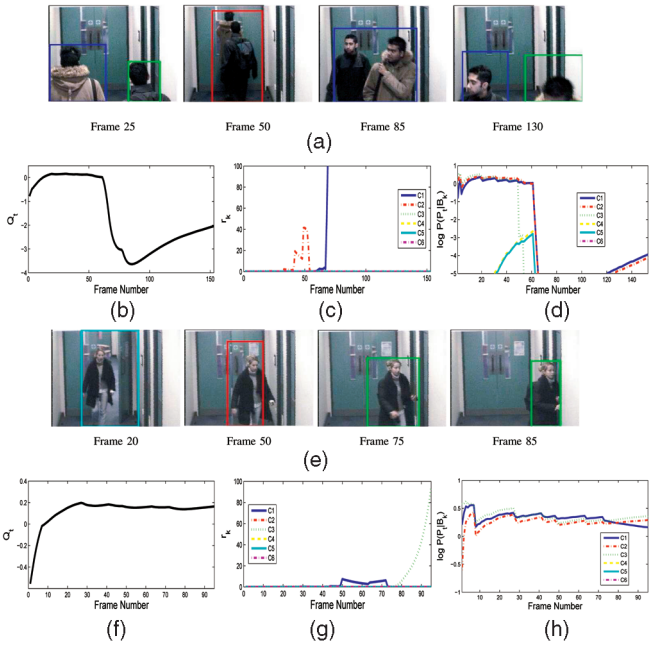


Fig. 9. Comparison of our LRT method with the ML method for online normal behavior recognition. (a) An abnormal behavior pattern where two people attempted to enter an office area without an entry card. It resembles C2 in the early stage. (b) The behavior pattern was detected as an anomaly from Frame 62 till the end based on Q_t. (c) The behavior pattern between Frames 40 and 53 was recognized reliably as C2 using our LRT method before being detected as an anomaly. (d) The behavior pattern was wrongly recognized as C3 before Frame 20 using the ML method. (e) A normal C3 behavior pattern. Note that it can be interpreted as either C1 or C3 before the person enters the side door. (f) The behavior pattern was detected as normal throughout using Q_t. (g) It was recognized reliably as C3 from Frame 83 till the end using our LRT method. (h) The behavior pattern was recognized prematurely and unreliably as either C1 or C3 before Frame 83 using the ML method.

6.2 Aircraft Docking Area Behavior Monitoring

Data set and feature extraction. Now, we consider an outdoor scenario. A fixed CCTV camera was mounted at an aircraft docking area, monitoring the aircraft docking procedure. Typical visually detectable behavior patterns in this scene involved the aircraft, the airbridge, and various ground vehicles (see Fig. 10). The captured video sequences have a very low frame rate of 2 Hz, which is common for CCTV surveillance videos. Each image frame has a size of 320×240 pixels. Our database for the experiments consists of 72,776 frames of video data (around 10 hours of recording) that cover different times of different days under changing

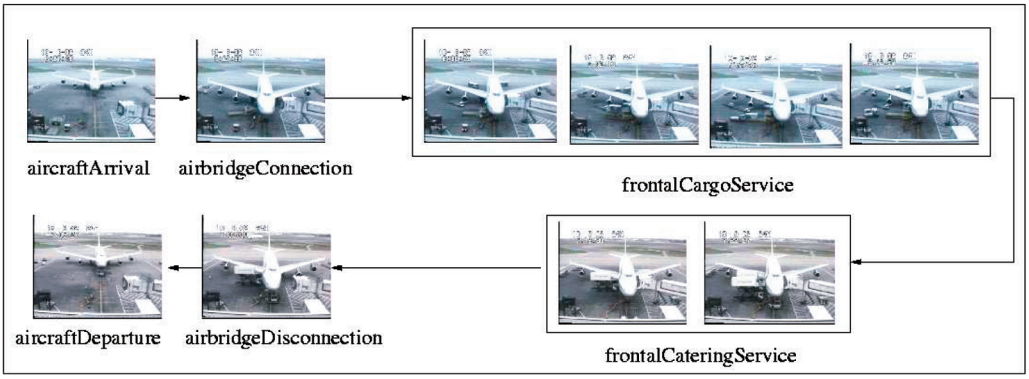
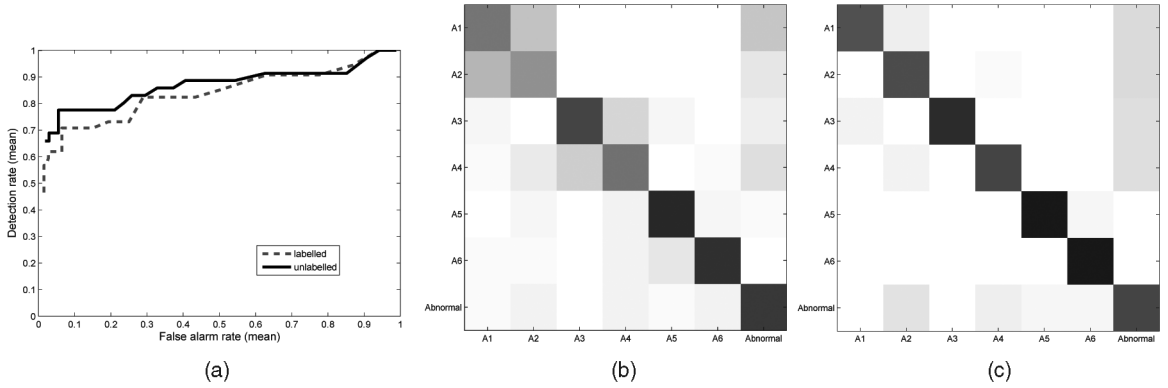


Fig. 10. Typical visually detectable behavior patterns in an aircraft docking scene.

TABLE 4
Six Classes of Commonly Occurred Behavior Patterns in the Airport Scene

A1	Aircraft arrives	A2	Airbridge connected	A3	Frontal cargo service
A4	Frontal catering service	A5	Aircraft departs	A6	Airbridge disconnected



(a) (b) (c)

Fig. 11. (a) Comparing the performance of anomaly detection using aircraft docking behavior models trained by labeled and unlabeled data. The mean ROC curves were obtained over 20 trials. (b) and (c) Comparing the performance of behavior recognition using models trained by labeled and unlabeled data. The two confusion matrices were obtained by averaging the results over 20 trials with $Th_A = -0.5$. Each row represents the probabilities of the corresponding class being confused with all the other classes averaged over 20 trials.

lighting conditions, from early morning to midday to late afternoon. The video was segmented automatically using an online segmentation algorithm proposed in [27], giving 59 video segments of actual behavior pattern instances. Each segment has on the average 428 frames, with the shortest having 74 and longest having 2,841.

Model training. A training set now consisted of 40 video segments, and the remaining 19 were used for testing. As in the corridor behavior monitoring experiments, 20 trials were conducted, each of which had a different random training set. To compare models trained using unlabeled data with those trained using labeled data, in each training session, a model was also trained using an identical but labeled training set. On the average, nine behavior classes were manually identified for the labeled training sets in each trial. Six classes were always identified in each training set (see Table 4). On the average, they accounted for 74 percent of the labeled training data.

Given a training set, discrete events were detected and classified, resulting in eight classes of events, corresponding to the common constituents of all behavior in this scene (see Fig. 1). These events were mainly triggered by the movements of the aircraft, airbridge, and various ground vehicles involved in an aircraft docking service circle. It is noted that our event detector makes more mistakes for the aircraft docking scene compared to that for the indoor corridor scene. This is due to the more challenging nature of the scenario in the sense that 1) the lighting condition in the aircraft scene was far less stable, 2) the image resolution of the moving objects in the aircraft scene were lower, and 3) the movements of different objects in the aircraft scene were occluded a lot more.

In each training session, the 40 unlabeled training behavior patterns were represented based on the event detection results and clustered to build a composite behavior model. On the average, seven eigenvectors were automatically determined as being relevant for clustering, with the smallest being 4 and largest being 10. The number of clusters for each training set was determined automatically as 2 and 6 in every trial without and with relevant eigenvector selection, respectively. By observation, each of the six discovered data clusters mainly contained samples corresponding to one of the six behavior classes listed in Table 4. In each session, a model was also built using labeled data for comparison.

Anomaly detection. Each behavior pattern in a testing set was detected as being abnormal when (17) was satisfied at any time after $T_w = 3K_e = 24$ frames. The accumulating factor α for computing Q_t was set to 0.1 (same as in the corridor behavior monitoring experiments). Fig. 11a shows that the models trained using unlabeled data outperformed those trained using labeled data (see also Table 5). The results are slightly inferior to those obtained in the corridor entrance/exit behavior modeling experiments (comparing Fig. 11a with Fig. 7a). It is not surprising since the data collected in this outdoor scenario are much more noisy and sparse, and the behavior captured in the data are more complicated. Figs. 12b and 12f show examples of online reliable anomaly detection using our runtime anomaly measure.

Recognition of normal behavior patterns. The normal behavior recognition results obtained using models trained by unlabeled and labeled data are illustrated in Table 6 and Figs. 11b and 11c. The results were consistent with those obtained in the corridor entrance/exit scene experiments. In particular, the recognition rate is slightly lower for models trained using unlabeled data. However, when not being recognized correctly, a normal behavior pattern is more likely to be detected as an anomaly using a labeled data

TABLE 5
The Mean and Standard Deviation of the Anomaly Detection Rate and False Alarm Rates for Aircraft Docking Behavior Models Trained Using Unlabeled and Labeled Data

	Anomaly Detection Rate (%)	False Alarm Rate (%)
Unlabelled	79.2 ± 8.3	5.1± 3.9
Labelled	71.0 ± 10.7	12.4 ± 5.2

The results were obtained over 20 trials with $Th_A = -0.5$.

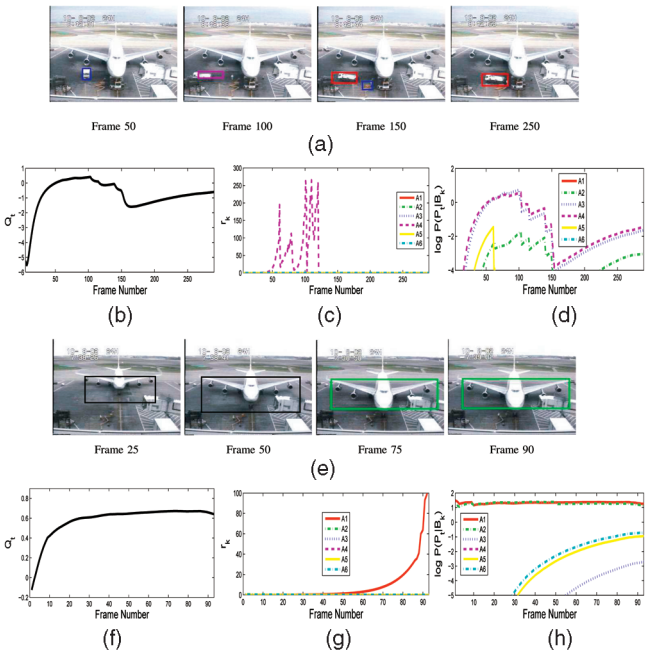


Fig. 12. Compare our LRT method with the ML method for online normal behavior recognition in an aircraft docking scene. (a) An abnormal behavior pattern where a truck brought engineers to fix a ground-power-cable problem. It resembled A4 in the early stage. (b) It was detected as an anomaly from Frame 147 till the end based on Q_t. (c) The behavior pattern between Frames 53 and 145 was recognized reliably as A4 using LRT before becoming abnormal and being detected using Q_t. (d) The behavior pattern was wrongly recognized as A3 between Frames 80 and 98 using the ML method. (e) A normal A1 behavior pattern. (f) The behavior pattern was detected as normal throughout based on Q_t. (g) It was recognized reliably as A1 from Frame 73 till the end using LRT. (h) It was wrongly recognized as A2 between Frames 12 and 49 using ML. In (a) and (e), detected events are illustrated using the same color scheme as in Fig. 1.

trained model. This resulted in a higher false alarm rate. Examples of online reliable behavior recognition using our online LRT method are shown in Fig. 12 in comparison with the ML method. It can be seen that our LRT method is superior to the ML method in that normal behavior patterns can be reliably and promptly recognized after sufficient visual evidence had become available to overcome the ambiguities among different behavior classes.

7 Discussions and Conclusions

The key findings of our experiments are summarized and discussed as follows:

1. Our experiments show that a behavior model trained using an unlabeled data set is superior to a model trained using the same but labeled data set in detecting anomaly from an unseen video. The former model also outperforms the latter in distinguishing abnormal behavior patterns from normal ones contaminated by errors in behavior representation. In comparison, a model trained using manually labeled data may have an advantage in explaining data that are well defined. However, training using labeled data does not necessarily help a model with identifying novel instances of abnormal behavior patterns as the model tends to be brittle and less robust in dealing with instances of behavior that are not clear cut in an open-world scenario (that is, the number of expected normal and abnormal behavior cannot be predefined exhaustively).
2. Our eigenvector selection-based spectral clustering algorithm is capable of discovering the natural grouping of behavior patterns in the training data. Our experiments show that our simple data-driven eigenvector selection algorithm works well on real-world behavior data. Furthermore, the eigenvector selection step of the algorithm is critical for determining the optimal number of behavior pattern classes.
3. Our online LRT-based normal behavior recognition method is superior to the conventional ML-based method. Since normal behavior recognition is performed after anomaly detection, it is plausible to construct the alternative model as a mixture of other normal behavior patterns. This is the key to make an LRT method work because the performance of an LRT method depends on the accuracy of constructing the alternative model.

TABLE 6
Comparing the Performance of Models Trained Using Unlabeled and Labeled Data on Aircraft Docking Normal Behavior Recognition

	Normal Behaviour Recognition Rate (%)
Unlabelled	72.1 ± 5.4
Labelled	79.5 ± 4.8

The results were obtained with $Th_A = -0.5$.

Since our event detection method is based on unsupervised learning, the detected event classes may not have a clear semantic meaning. In addition, some event classes deemed as important by human may not be detected due to the lack of visual evidence or ambiguities between event classes (for example, the "swipe card" event in the corridor entrance/exit scene). Nevertheless, our experiments suggest that the proposed method is able to cope with the errors in event detection. This, as we mentioned earlier, is mainly due to the fact that the temporal information about the occurrence of events is exploited under a probabilistic framework.

It is possible to develop a Baum-Welch-like [1] EM algorithm to the mixture of DBNs (14) to learn the behavior model directly rather than taking a stepped approach as proposed in the paper. A model selection criterion such as BIC can then be employed to determine the number of behavior classes automatically, which corresponds to the number of mixtures in the directly learned model. However, in practice, learning the model directly brings about the initialization problem, which could lead to poorer results compared to the proposed stepped approach. The initialization problem always exists for a model-based clustering algorithm (for example, GMMs). However, the problem becomes rather acute in the case of clustering using a mixture of DBNs because the number of parameters needed to describe the model is very large and the EM algorithm used for model learning will suffer severely from the local minimum problem and prone to overfitting. It would be interesting to investigate possible solutions to the initialization problem and compare the results obtained using the direct approach with those of the stepped approach proposed in this work.

It is also noted that our behavior profiling framework is flexible in the sense that it allows for the use of different behavior segmentation, representation, and affinity

measurement as long as a behavior affinity matrix can be constructed. Furthermore, the unsupervised nature of the framework allows it to be quickly adaptive to different surveillance scenarios.

In conclusion, we have proposed a novel framework for robust online behavior recognition and anomaly detection. The framework is fully unsupervised and consisted of a number of key components, namely, a discrete event-based behavior representation, a DBN-based behavior affinity measure, a spectral clustering algorithm with feature and model selection, a composite behavior model based on a mixture of DBNs, a runtime accumulative anomaly measure, and an online LRT-based normal behavior recognition method. The effectiveness and robustness of our approach is demonstrated through experiments using data sets collected from both indoor and outdoor surveillance scenarios.

This work is a serious effort to address the problem of anomaly detection in realistic scenarios. Nevertheless, there is still a long way to go toward a general-purpose anomaly detection method that can be applied to any type of scenarios. In particular, the proposed approach will not be able to cope with a very busy and unstructured scenario. The limitation is mainly caused by the representation aspect of the approach. More specifically, extremely crowded dynamic scenes cause problem to our event detection method because it is very difficult and ambiguous to define and measure significant visual changes that should be associated to events. One of the possible improvements we can make would be developing more sophisticated and robust event detection methods based on the analysis of flow dynamics for extremely crowded dynamic scenes and investigating the possibility of combining rule-based behavior profiling approaches with statistical learning-based approaches. Another drawback of the work is that it does not cope with changes of visual context that could affect the definition of what is normal and abnormal. In other words, once trained, the model cannot be adapted automatically to new observations. Our ongoing work therefore also includes adding adaptive and incremental learning features into the framework.

Finally, it is worth pointing out that despite the best efforts from an increasing number of researchers, we are still at the very early stage of understanding the video-behavior-anomaly detection problem. We believe strongly that in order to make significant progress, a number of open questions need to be addressed. One of the questions is about how to make use of domain knowledge for anomaly detection. One of reasons why human can detect anomaly so effortlessly is because an enormous amount of domain knowledge/common sense about the scenarios has been accumulated and employed for decision making. How can a machine learn such knowledge, which is hidden in an almost unlimited amount of data? How can the learning process be speeded up with supervisions from human? A closely related question would be how knowledge about one domain can be generalized to others? Again, human has the ability to generalize knowledge so that anomaly can be detected even in an unfamiliar domain. How to enable a machine to possess the same ability so that we do not have to redo the learning from scratch when an anomaly detection algorithm is applied to a completely different scenario? We envisage that these questions will be the focus of research on video anomaly detection for years to come.

REFERENCES

[1] L.E. Baum and T. Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains," *Annals Math. Statistics,* vol. 37, pp. 1554-1563, 1966.

[2] O. Boiman and M. Irani, "Detecting Irregularities in Images and in Video," *Proc. 10th IEEE Int'l Conf. Computer Vision,* pp. 462-469, 2005.

[3] H. Dee and D. Hogg, "Detecting Inexplicable Behaviour," *Proc. British Machine Vision Conf.,* pp. 477-486, 2004.

[4] A. Dempster, N. Laird, and D. Rubin, "Maximum-Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc. B,* vol. 39, pp. 1-38, 1977.

[5] T. Duong, H. Bui, D. Phung, and S. Venkatesh, "Activity Recognition and Abnormality Detection with the Switching Hidden Semi-Markov Model," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 838-845, 2005.

[6] J. Dy, C. Brodley, A. Kak, L. Broderick, and A. Aisen, "Unsupervised Feature Selection Applied to Content-Based Retrieval of Lung Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, pp. 373-378, 2003.

[7] Z. Ghahramani, "Learning Dynamic Bayesian Networks," *Adaptive Processing of Sequences and Data Structures: Int'l Summer School on Neural Networks,* pp. 168-197, 1998.

[8] S. Gong and T. Xiang, "Recognition of Group Activities Using Dynamic Probabilistic Networks," *Proc. Ninth IEEE Int'l Conf. Computer Vision,* pp. 742-749, 2003.

[9] R. Hamid, A. Johnson, S. Batta, A. Bobick, C. Isbell, and G. Coleman, "Detection and Explanation of Anomalous Activities: Representing Activities as Bags of Event N-Grams," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1031-1038, 2005.

[10] A. Higgins, L. Bahler, and J. Porter, "Speaker Verification Using Randomized Phrase Prompting," *Digital Signal Processing,* pp. 89-106, 1991.

[11] J. Kruskal and M. Liberman, *The Symmetric Time-Warping Problem: From Continuous to Discrete.* Addison-Wesley, 1983.

[12] M. Law, M.A.T. Figueiredo, and A.K. Jain, "Simultaneous Feature Selection and Clustering Using Mixture Model," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 9, pp. 1154-1166, Sept. 2004.

[13] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE,* vol. 77, no. 2, pp. 257-286, 1989.

[14] R.J. Morris and D.C. Hogg, "Statistical Models of Object Interaction," *Int'l J. Computer Vision,* vol. 37, no. 2, pp. 209-215, 2000.

[15] A. Ng, M. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," *Advances in Neural Information Processing Systems,* 2001.

[16] N. Oliver, B. Rosario, and A. Pentland, "A Bayesian Computer Vision System for Modelling Human Interactions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 831-843, Aug. 2000.

[17] A. Panuccio, M. Bicego, and V. Murino, "A Hidden Markov Model-Based Approach to Sequential Data Clustering," *Proc. Joint IAPR Int'l Workshop Structural, Syntactic, and Statistical Pattern Recognition,* pp. 734-742, 2002.

[18] F. Porikli, "Trajectory Distance Metric Using Hidden Markov Model Based Representation," *Proc. Sixth IEEE Int'l Workshop Performance Evaluation of Tracking and Surveillance,* 2002.

[19] F. Porikli and T. Haga, "Event Detection by Eigenvector Decomposition Using Object and Frame Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshop,* pp. 114-121, 2004.

[20] S. Roberts, D. Husmeier, I. Rezek, and W. Penny, "Bayesian Approaches to Gaussian Mixture Modeling," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 11, pp. 1133-1142, Nov. 1998.

[21] G. Schwarz, "Estimating the Dimension of a Model," *Annals Statistics,* vol. 6, pp. 461-464, 1978.

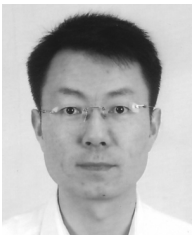
[22] Y. Shan, H.S. Sawhney, and A. Pope, "Measuring the Similarity of Two Image Sequence," *Proc. Asian Conf. Computer Vision,* 2004.

[23] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.

[24] C. Stauffer and W. Grimson, "Learning Patterns of Activity Using Real-Time Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 747-758, Aug. 2000.

[25] Y. Weiss, "Segmentation Using Eigenvectors: A Unifying View," *Proc. Seventh IEEE Int'l Conf. Computer Vision,* pp. 975-982, 1999.

[26] J. Wilpon, L. Rabiner, C. Lee, and E. Goldman, "Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models," *IEEE Trans. Acoustics, Speech, and Signal Processing*, pp. 1870-1878, 1990.
[27] T. Xiang and S. Gong, "Activity Based Video Content Trajectory Representation and Segmentation," *Proc. British Machine Vision Conf.*, pp. 177-186, 2004.
[28] T. Xiang and S. Gong, "Video Behaviour Profiling and Abnormality Detection without Manual Labelling," *Proc. 10th IEEE Int'l Conf. Computer Vision*, pp. 1238-1245, 2005.
[29] T. Xiang, S. Gong, and D. Parkinson, "Autonomous Visual Events Detection and Classification without Explicit Object-Centred Segmentation and Tracking," *Proc. British Machine Vision Conf.*, pp. 233-242, 2002.
[30] S. Yu and J. Shi, "Multiclass Spectral Clustering," *Proc. Ninth IEEE Int'l Conf. Computer Vision*, pp. 313-319, 2003.
[31] L. Zelnik-Manor and M. Irani, "Event-Based Video Analysis," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2001.
[32] L. Zelnik-Manor and P. Perona, "Self-Tuning Spectral Clustering," *Advances in Neural Information Processing Systems*, 2004.
[33] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan, "Semi-Supervised Adapted HMMS for Unusual Event Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 611-618, 2005.
[34] H. Zhong, J. Shi, and M. Visontai, "Detecting Unusual Activity in Video," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 819-826, 2004.



Tao Xiang received the BS degree in electrical engineering from Xi'an Jiaotong University in 1995, the MS degree in electrical engineering from the Communication University of China in 1998, and the PhD degree in electrical and computer engineering from the National University of Singapore in 2002. He is a currently a lecturer in the Department of Computer Science, Queen Mary, University of London. His research interests include computer vision, statistical learning, video processing, and machine learning, with focus on interpreting and understanding human behavior. He is a member of the IEEE.

Shaogang Gong received the DPhil degree in computer vision from Oxford University in 1989, with a thesis on the computation of optic flow using second-order geometric analysis. He is a professor of visual computation at Queen Mary, University of London. He heads both the Queen Mary Computer Vision Research Group and the Queen Mary Computer Science, Mathematics, and Electronic Engineering Research Consortium on Digital Media and Complexity Sciences. He has published more than 170 papers on computer vision and machine learning and the book *Dynamic Vision: From Images to Face Recognition*. His work focuses on motion and video analysis, object detection, tracking and recognition, face and expression recognition, gesture and action recognition, human behavior recognition, and anomaly detection. He is an elected fellow of the IEE and a member of the UK Computing Research Committee. He was a recipient of a Queen's Research Scientist Award in 1987, a Royal Society Research Fellow in 1987 and 1988, and a GEC-Oxford Fellow in 1989. He twice received the Best Science Prize of the British Machine Vision Conferences (BMVC '99 and '01) and received the Best Paper Award of the IEEE International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures (RATFG '01) and the Best Paper Award of the IEE International Symposium on Imaging for Crime Detection and Prevention (ICDP '05).