Video Behaviour Profiling for Anomaly Detection

Tao Xiang[1] and Shaogang Gong[2]

Department of Computer Science

Queen Mary, University of London, London E1 4NS, UK

Abstract

This paper aims to address the problem of modelling video behaviour captured in surveillance videos for the applications of online normal behaviour recognition and anomaly detection. A novel framework is developed for automatic behaviour profiling and online anomaly sampling/detection without any manual labelling of the training dataset. The framework consists of the following key components: (1) A compact and effective behaviour representation method is developed based on discrete scene event detection. The similarity between behaviour patterns are measured based on modelling each pattern using a Dynamic Bayesian Network (DBN). (2) Natural grouping of behaviour patterns is discovered through a novel spectral clustering algorithm with unsupervised model selection and feature selection on the eigenvectors of a normalised affinity matrix. (3) A composite generative behaviour model is constructed which is capable of generalising from a small training set to accommodate variations in unseen normal behaviour patterns. (4) A run-time accumulative anomaly measure is introduced to detect abnormal behaviour while normal behaviour patterns are recognised 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 behaviour recognition at the shortest possible time. The effectiveness and robustness of our approach is demonstrated through experiments using noisy and sparse datasets collected from both indoor and outdoor surveillance scenarios. In particular, it is shown that a behaviour model trained using an unlabelled dataset is superior to those trained using the same but labelled dataset in detecting anomaly from an unseen video. The experiments also suggest that our online LRT based behaviour recognition approach is advantageous over the commonly used *Maximum Likelihood* (ML) method in differentiating ambiguities among different behaviour classes observed online.

Index Terms

Behaviour profiling, Anomaly Detection, Dynamic Scene Modelling, Spectral clustering, Feature Selection, Dynamic Bayesian Networks.

[1] Corresponding author. Tel:(+44)-(0)20-7882-5201; Fax: (+44)-(0)20-8980-6533; Email: txiang@dcs.qmul.ac.uk

[2] Email: sgg@dcs.qmul.ac.uk

I. INTRODUCTION

There is an increasing demand for automatic methods for analysing the vast quantities of surveillance video data generated continuously by CCTV systems. One of the key objectives of deploying an automated visual surveillance system is to detect abnormal behaviour patterns and recognise the normal ones. To achieve this objective previously observed behaviour patterns need to be analysed 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 analysed and the real-time nature of many surveillance applications, it is very desirable to have an automated system which runs in real-time and requires little human intervention. In the paper, we aim to develop such a system which is based on fully unsupervised behaviour profiling and robust online anomaly detection.

Let us first define the problem of automatic behaviour profiling for anomaly detection. Given 24/7 continuously recorded video or online CCTV input, the goal of automatic behaviour profiling is to learn a model that is capable of detecting unseen abnormal behaviour patterns whilst recognising novel instances of expected normal behaviour patterns. In this context, we define an anomaly as an atypical behaviour pattern that is not represented by sufficient samples in a training dataset but critically it 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 behaviour representation. The effectiveness of a behaviour profiling algorithm shall be measured by (1) how well anomalies can be detected (i.e. measuring specificity to expected patterns of behaviour) and (2) how accurately and robustly different classes of normal behaviour patterns can be recognised (i.e. maximising between-class discrimination).

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

1) A scene event based behaviour representation. Due to the space-time nature of behaviour patterns and their variable durations, we need to develop a compact and effective behaviour 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 behaviour representation aims to avoid the difficulties associated with tracking under occlusion in noisy scenes [8]. Each behaviour pattern is modelled using a Dynamic Bayesian Network (DBN) [7] which provides a suitable means for time-warping and measuring the affinity between

behaviour patterns.

2) Behaviour profiling based on discovering natural grouping of behaviour patterns using the relevant eigenvectors of a normalised behaviour affinity matrix. A number of affinity matrix based clustering techniques have been proposed recently [25], [23], [30]. However, these approaches require known number of clusters. Given an unlabelled dataset, the number of behaviour classes are unknown in our case. To automatically determine the number of clusters, we propose to first perform unsupervised feature selection to eliminate those eigenvectors that are irrelevant/uninformative in behaviour 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 behaviour model using a mixture of DBNs. The advantages of the such a generative behaviour model are two fold: (a) It can accommodate well the variations in the unseen normal behaviour patterns both in terms of duration and temporal ordering by generalising from a training set of limited number of samples. This is important because in reality the same normal behaviour can be executed in many different normal ways. These variations cannot possibly be captured in a limited training dataset and need to be dealt with by a learned behaviour model. (b) Such a model is robust to errors in behaviour representation. A mixture of DBNs can cope with errors occurred at individual frames and is also able to distinguish an error corrupted normal behaviour pattern from an abnormal one.
4) Online anomaly detection using a run-time accumulative anomaly measure and normal behaviour recognition using an online Likelihood Ratio Test (LRT) method. A run-time accumulative measure is introduced to determine how normal/abnormal an unseen behaviour pattern is on-the-fly. The behaviour pattern is then recognised as one of the normal behaviour classes if detected as being normal. Normal behaviour 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 behaviour classes observed online due to insufficient visual evidence at a given time instance. By doing so, robust behaviour recognition and anomaly detection are ensured at the shortest possible time, as opposed to previous work such as [2], [8], [16] which requires completed behaviour patterns being observed. Our online LRT based behaviour recognition approach is also advantageous over previous

ones based on the *Maximum Likelihood* (ML) method [31], [8], [16]. A ML based approach makes a forced decision on behaviour 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 labelling is avoided in both feature extraction for behaviour representation and discovery of natural grouping of behaviour patterns. There are a number of motivations for performing behaviour profiling using unlabelled data: first, manual labelling of behaviour patterns is laborious and often rendered impractical given the vast amount of surveillance video data to be processed. More critically though, manual labelling of behaviour patterns could be inconsistent and error prone. This is because a human tends to interpret behaviour based on *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 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 VI, is able to cope with a moderately crowded scenario thanks to the discrete event based behaviour 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 II reviews related work to highlight the contributions of this work. Section III addresses the problem of behaviour representation. The behaviour profiling process is described in Section IV, which also explains how a composite generative behaviour model can be built using a mixture of DBNs. Section V centres about online detection of abnormal behaviour and recognition of normal behaviour using a behaviour model. In Section VI, the effectiveness and robustness of our approach is demonstrated through experiments using noisy and sparse datasets collected from both indoor and outdoor surveillance scenarios. The paper concludes in Section VII.

II. Related Work

Much work on abnormal behaviour 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* behaviour classes (both normal and abnormal). However, in reality abnormal behaviour is both

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

rare and far from being well-defined, resulting in insufficient clearly labelled data required for supervised model building.

More recently, a number of techniques have been proposed for unsupervised learning of behaviour models [34], [9], [2], [28]. They can be further categorised into two different types according to whether an explicit model is built. Approaches that do not model behaviour explicitly either perform clustering on observed patterns and label those forming small clusters as being abnormal [34], [9] or build a database of spatio-temoral patches using only regular/normal behaviour 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 behaviour patterns 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 behaviour patterns must be stored either in the form of histograms of discrete events [9] or ensembles of spatio-temporal patches [2] for detecting anomaly from unseen data, which jeopardises the scalability of these approaches.

There is also another approach that differs from both the supervised and unsupervised techniques above. A semi-supervised model was introduced by [33] with a two-stages training process. In stage one, a normal behaviour model is learned using labelled normal patterns. In stage two, an abnormal behaviour model is then learned unsupervised using Bayesian adaptation. This approach still suffers from the laborious and inconsistent manual data labelling process.

In our work, an explicit model based on a mixture of Dynamic Bayesian Networks (DBNs) is constructed in an unsupervised manner to learn specific behaviour 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 behaviour recognition based on a run-time accumulative anomaly measure and an online Likelihood Ratio Test (LRT) method originally proposed for key-words detection in speech recognition [26]. This makes our approach more robust to noise in behaviour representation. Our approach is similar to [9] in that behaviour patterns are represented using discrete events. However, manual labelling of objects of interests and manual event annotation are required in [9] which make it not fully unsupervised. Moreover, the behaviour representation in [9] is based on event histogram that ignores/throws away any information about the duration of an event. Such a behaviour 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 behaviour pattern is observed, whist in our work it is performed on-the-fly. Our work is similar in spirit to [2] in that the behaviour model (constructed in [2] as a database of video patches) is able to infer and generalise 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 behaviour pattern due to the constraint on the size of the video patches. In particular, the approach can only detect unusual local spatio-temporal formations from a single objects rather than subtle abnormalities embedded in the temporal correlations among multiple objects which are not necessarily close to each other in space and time. In summary, the key advantages of the proposed approach over previous approaches are: (1) It is based on constructing a composite generative behaviour model which scales well with the complexity of behaviour and is robust to errors in behaviour representation; (2) it performs on-the-fly anomaly detection and is therefore suitable for real-time surveillance applications.

III. Behaviour Pattern Representation

A. Video Segmentation

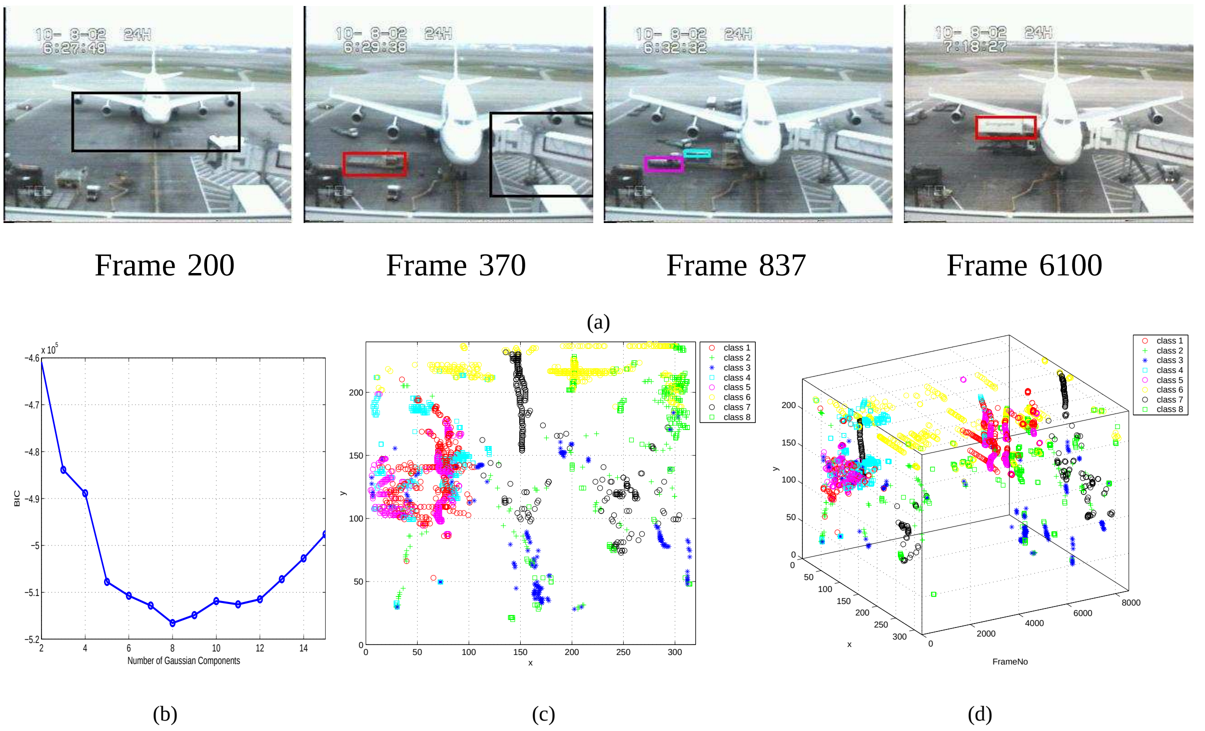
The goal is to automatically segment a continuous video sequence $\mathbf{V}$ into N video segments $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_n, \ldots, \mathbf{v}_N\}$ such that ideally each segment contains a single behaviour pattern. The nth video segment $\mathbf{v}_n$ consisting of T_n image 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 non-activity gaps between two consecutive behaviour patterns which can be utilised for activity segmentation. In the case where obvious non-activity gaps are not available, an on-line 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 on-line using a Forward-Backward Relevance (FBR) procedure. Alternatively, the video can be simply sliced into overlapping segments with a fixed time duration [34].

B. Event-based Behaviour Representation

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

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

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_px, M_py) 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_px, M_py) are motion features capturing the direction of object motion.



Frame 200 Frame 370 Frame 837 Frame 6100

(a)

(b) (c) (d)

Fig. 1. Event-based behaviour representation for an aircraft docking video. Details of the video can be found at Section VI-B. (b) shows that 8 classes of events are detected automatically using BIC. Different classes of events are highlighted in the image frame using bounding boxes in different colours in (a). Spatial and temporal distribution of events of different classes throughout the sequence are 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 moving and stopping aircraft and airbridges are indicated in black and green (rectangular) respectively; Events corresponding to movements of aircraft-pushing vehicles, passing-by vehicles in the back, and rear catering vehicles are in blue, yellow, and green (cross) respectively.

Thirdly, 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 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 behaviour pattern captured in the nth video segment $\mathbf{v}_n$ is represented as a feature vector $\mathbf{P}_n$, given as

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

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:

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

$\mathbf{p}_{nt}$ 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 behaviour representation is illustrated through an example in Fig. 1. Note that different classes of events occurred simultaneously (see Fig. 1(a)).

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

IV. Behaviour Profiling

The behaviour profiling problem can now be defined formally. Consider a training dataset $\mathbf{D}$ 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 Eqn. (2) representing the behaviour pattern captured by the nth video segment $\mathbf{v}_n$. The problem to be addressed is to discover the natural grouping of the

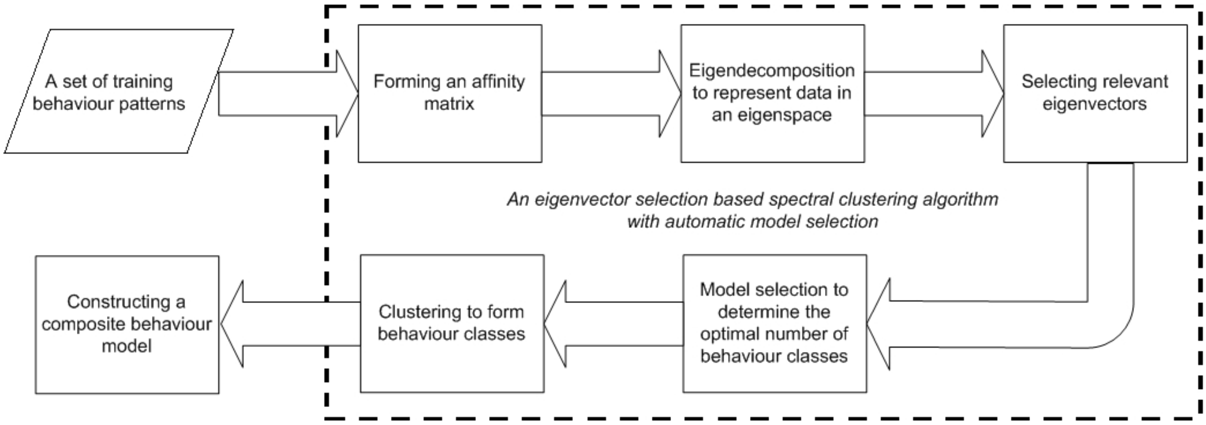

A set of training behaviour patterns
Forming an affinity matrix
Eigendecomposition to represent data in an eigenspace
Selecting relevant eigenvectors
An eigenvector selection based spectral clustering algorithm with automatic model selection
Constructing a composite behaviour model
Clustering to form behaviour classes
Model selection to determine the optimal number of behaviour classes

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

training behaviour patterns upon which a model for normal behaviour 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 [11]. A standard dynamic time warping (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 behaviour patterns by finding correspondence between discrete vertices on two trajectories. Since in our framework, a behaviour pattern is represented as a set of temporal correlated events, i.e. a stochastic process, a stochastic modelling 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 modelling behaviour that could involve multiple objects interacting over space and time, the approach in [22] can not be applied directly in our case. (3) Model selection needs to be performed to determine the number of clusters. To overcome the above-mentioned difficulties, we propose a spectral clustering algorithm with feature and model selection based on modelling each behaviour pattern using a Dynamic Bayesian Network (DBN). Fig. 2 shows a diagrammatic illustration of our behaviour 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 subsections

A. Affinity Matrix

Dynamic Bayesian Networks (DBNs) provide a solution for measuring the affinity between different behaviour patterns. More specifically, each behaviour pattern in the training set is modelled using a DBN. To measure the affinity between two behaviour 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 EM algorithm [4], [7]. The affinity between $\mathbf{P}_i$ and $\mathbf{P}_j$ is then computed as:

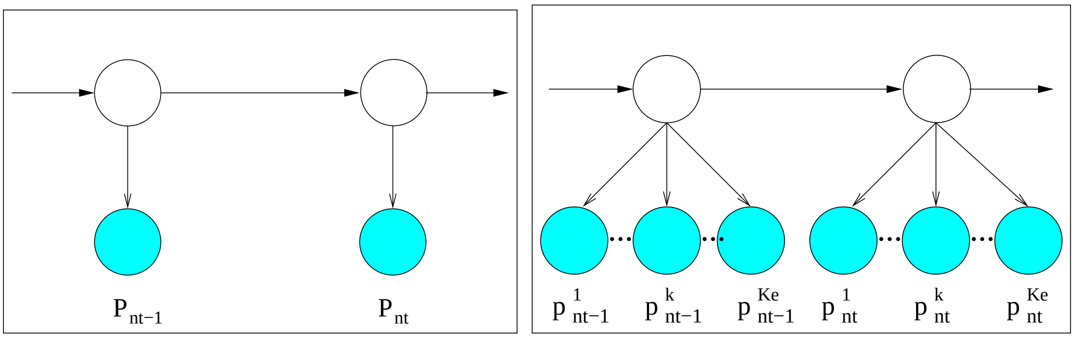
$$S_{ij} = \frac{1}{2}\left\{\frac{1}{T_j}\log P(\mathbf{P}_j|\mathbf{B}_i) + \frac{1}{T_i}\log P(\mathbf{P}_i|\mathbf{B}_j)\right\}, \tag{5}$$

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. 3(a)). In this HMM, the observation variable at each time instance corresponds to $\mathbf{p}_{nt}$ (Eqn. (3)), which represents the content of the tth frame of the nth behaviour pattern, and is of dimension K_e, i.e. the number of event classes. The conditional probability distributions (CPD) of $\mathbf{p}_{nt}$ is assumed to be Gaussian for each of the N_s states of its parent node. However, a drawback of a HMM is that too many parameters are needed to describe the model when the observation variables are of high dimension. This makes a HMM vulnerable to overfitting therefore generating poorly to unseen data. It is especially true in our case because a HMM needs to be learned for every single behaviour pattern in the training dataset which could be short in duration. To solve this problem, we employ a Multi-Observation Hidden Markov Model (MOHMM) [8] shown in Fig. 3(b). Compared to a HMM, the observational space is factorised 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 a 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 a HMM). In this paper, N_s, the number of hidden states for each hidden variables in the MOHMM, is set to K_e, i.e., the number of event classes. This is reasonable because the value of N_s should reflect the complexity of a behaviour pattern, so should the value of K_e.

An $N \times N$ affinity matrix $\mathbf{S} = [S_{ij}]$ where $1 \leq i, j \leq N$ provides a new representation for the training dataset, denoted as $\mathbf{D_s}$. In this representation, a behaviour pattern is represented

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



(a) HMM

(b) MOHMM

Fig. 3. Modelling a behaviour pattern $\mathbf{P}_n = \{\mathbf{p}_{n1}, \ldots, \mathbf{p}_{nt}, \ldots, \mathbf{p}_{nT_n}\}$ where $\mathbf{p}_{nt} = \{p^1_{nt}, ..., p^k_{nt}, ..., p^{K_e}_{nt}\}$ using a HMM and a MOHMM. Observation nodes are shown as shaded circles and hidden nodes as clear circles.

by its affinity to each behaviour pattern in the training set. Specifically, the nth behaviour pattern is now represented as the nth row of $\mathbf{S}$, denoted as $\mathbf{s}_n$. We thus have

$$\mathbf{D_s} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n, \ldots, \mathbf{s}_N\}. \tag{6}$$

Consequently each behaviour pattern is represented as a feature vector of a fixed length N. Taking a conventional data clustering approach, model selection can be performed firstly 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 Neighbour 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 which reduces the data dimensionality and performs clustering using the selected relevant eigenvectors of the data affinity matrix.

B. Eigendecomposition

Dimension reduction on the N dimensional feature space defined in Eqn. (6) can be achieved through 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 normalised affinity matrix $\bar{\mathbf{S}}$, defined as:

$$\bar{\mathbf{S}} = \mathbf{L}^{-\frac{1}{2}}\mathbf{S}\mathbf{L}^{-\frac{1}{2}} \tag{7}$$

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 $\bar{\mathbf{S}}$ (i.e. eigenvectors with

[3] The largest eigenvectors are eigenvectors that their corresponding eigenvalues are the largest in magnitude.

the largest eigenvalues) are sufficient to partition the dataset into K clusters. Representing the dataset using the K largest eigenvectors reduces the data dimensionality from N (i.e. the number of behaviour patterns) to K (i.e. the number of behaviour 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 the K, which is unknown. This is solved through automatic model selection.

C. 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 dataset. This is a reasonable assumption because as a rule of thumb a more sparse dataset (i.e. $\frac{1}{5}N < K < K_m$) would make any data clustering algorithm unworkable. The training dataset is now 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 behaviour 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 Gaussian Mixture Model (GMM). The log-likelihood of observing the training dataset $\mathbf{D_e}$ 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 Bayesian Information Criterion (BIC) is then employed to select the optimal number of components K determining the number of behaviour 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. 4(g) for an example and more in Section VI). This is not surprising because BIC has been known for having the tendency of underfitting a model given sparse data [20]. A dataset 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, i.e. selecting relevant/informative eigenvectors of the normalised affinity matrix to perform model selection and data clustering.

D. Eigenvector Selection for Behaviour Pattern Clustering

We aim to derive a suitable criterion for measuring the relevance of each eigenvector of the normalised 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 investigating 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, 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, i.e., 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 dataset into different clusters. This is achieved through modelling the distributions of the elements of each eigenvectors 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 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:

$$p(e_{kn}|\theta_{e_{kn}}) = (1 - R_{\mathbf{e_k}})p\left(e_{kn}|\theta^1_{e_{kn}}\right) + R_{\mathbf{e_k}}p\left(e_{kn}|\theta^2_{e_{kn}}\right)$$

where $\theta_{e_{kn}}$ are the parameters describing the distribution, $p(e_{kn}|\theta^1_{e_{kn}})$ 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 ($p(e_{kn}|\theta^1_{e_{kn}})$); (2) its simple form means that only small number of parameters are needed to describe $p(e_{kn}|\theta^2_{e_{kn}})$. This makes model learning possible even given sparse data.

There are 8 parameters required for describing the distribution of e_{kn}:

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

The maximum likelihood (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 rest 6 parameters are then estimated iteratively using Expectation Maximisation (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 8 parameters are re-estimated in each iteration, the distribution of e_{kn} is essentially modelled 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, i.e. 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 initialisation especially 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 utilised 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 normalised eigenvalue for $\mathbf{e_k}$ with $\bar{\lambda}_1 = 1$ and $\bar{\lambda}_{K_m} = 0$. We then randomly initialise the values of the other five parameters, namely $\mu_{k2}, \mu_{k3}, \sigma_{k2}, \sigma_{k3}$ and w_k, and the solution that yields the largest $p(e_{kn}|\theta^2_{e_{kn}})$ over different initialisations is chosen.

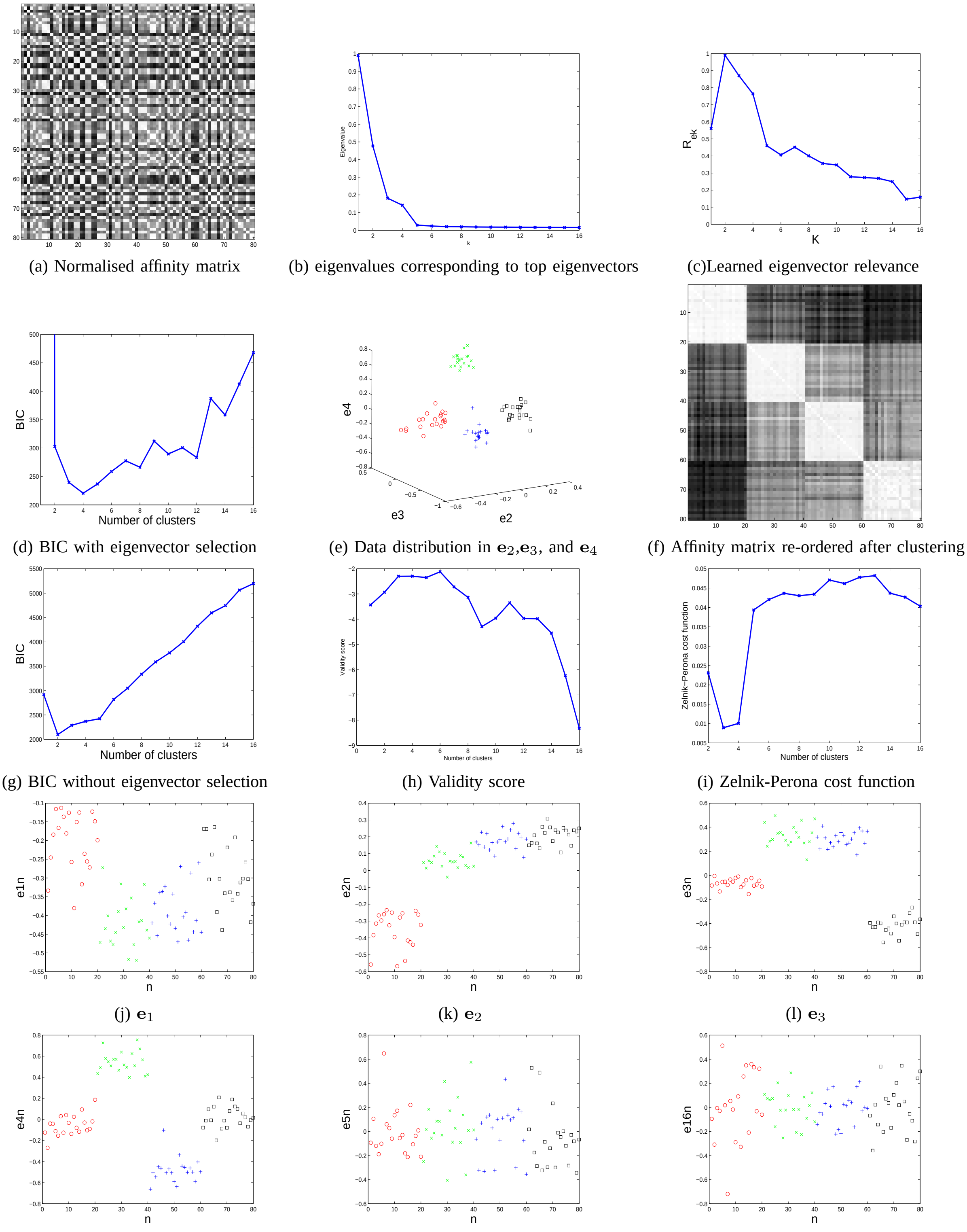
It is worth pointing out that although our relevance learning algorithm is based on estimating the distribution of the elements of each eigenvector, we are only interested in learning whether the distribution is unimodel or multimodal, which is reflected by the value of $R_{\mathbf{e_k}}$. In other words, among the 8 free parameters of the eigenvector distribution (Eqn. (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 2 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

$$\hat{R_{\mathbf{e_k}}} < 0.5 \tag{13}$$

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 IV-C. Each behaviour pattern in the training dataset is then labelled as one of the K behaviour classes.

Fig. 4 shows an example of data clustering using our eigenvector selection based spectral clustering algorithm. 80 sequences were randomly generated using four MOHMMs. The dataset 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 shown in Fig. 3(b) with different parameters set randomly. It can be seen from Fig. 4(j)-(o) that the second, third, and fourth eigenvectors contain strong information about the grouping of data while the largest eigenvector is much less informative. The rest eigenvectors contain virtually no information (see Fig. 4(n)&(o)). It can be seen form Fig. 4(c) that the proposed relevance measure $R_{\mathbf{e_k}}$ accurately reflects the relevance of each eigenvectors. By thresholding the relevance measure (Eqn. (13)), the largest 4 eigenvectors are kept for clustering. Fig. 4(e) shows that the 4 clusters are clearly separable in the eigenspace spanning the 3 most relevant eigenvectors. It is thus not surprising that the number of clusters was determined correctly as 4 using BIC on the relevant eigenvectors (see Fig. 4(d)). The



(a) Normalised 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 re-ordered 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) $\mathbf{e}_{16}$

Fig. 4. Clustering a synthetic dataset 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 Eqn. (13). (d) shows the BIC model selection results; the optimal cluster number was determined as 4. (e): the 80 data sample plotted using the three most relevant eigenvectors, i.e. $\mathbf{e}_2$,$\mathbf{e}_3$, and $\mathbf{e}_4$. Points corresponding to different classes are colour coded in (e) according to the classification result. (f): the affinity matrix re-ordered according to the result of our clustering algorithm. (g)-(i) show the cluster number was estimated as 2, 5, and 3 respectively using three alternative approaches. The distribution of some of the top 16 eigenvectors are shown in (j)-(o).

clustering result is illustrated using the re-ordered affinity matrix in Fig. 4(f) showing that all four clusters were discovered accurately. We also estimated the number of clusters using three alternative methods: (a) BIC using all 16 eigenvectors; (b) Porikli and Haga's Validity score [19] (maximum score correspond to the optimal number); and (c) Zelnik-Perona cost function [32] (minimum cost correspond to the optimal number). Fig. 4(g)-(i) show that none of these methods was able yield an accurate estimate of the cluster number.

E. A Composite Behaviour Model using a Mixture of MOHMMs

To build a model for the observed/expected behaviour, we first model the kth behaviour 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 behaviour model $\mathbf{M}$ is then formulated as a mixture of the K MOHMMs. Given an unseen behaviour pattern, represented as a behaviour pattern feature vector $\mathbf{P}$ as described in Section III, 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 behaviour patterns and N_k is the number of patterns that belong to the kth behaviour class.

V. Online Anomaly Detection and Normal Behaviour Recognition

Once constructed, the composite behaviour model $\mathbf{M}$ can be used to detect whether an unseen behaviour pattern is normal using a run-time anomaly measure. If detected as being normal, the behaviour pattern is then recognised as one of the K classes of normal behaviour patterns using an online Likelihood Ratio Test (LRT) method.

An unseen behaviour 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 behaviour pattern, represented as $\mathbf{P}_t = [\mathbf{p}_1, \ldots, \mathbf{p}_t]$, is used for online reliable anomaly detection. First the normalised log-likelihood of observing $\mathbf{P}$ at the tth frame given the behaviour 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 rate required by each particular surveillance application. Note that it takes a time delay for Q_t to stabilise at the beginning of evaluating a behaviour 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 behaviour modelling. We thus set $T_w = 3K_e$ in our experiments to be reported later in Section VI, i.e. the anomaly of a behaviour pattern is only evaluated when $t > T_w$.

At each frame t a behaviour pattern needs to be recognised as one of the K behaviour classes when it is detected as being normal, i.e. $Q_t > Th_A$. This is achieved using an online Likelihood Ratio Test (LRT) method. More specifically, we consider a hypotheses test between:

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

where H_0 is called the alternative hypothesis. Using LRT, we compute the likelihood ratio of the 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 behaviour profiling step. The key to LRT is thus to construct the alternative model which 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 behaviour classes. Therefore it is reasonable to construct the alternative model as a mixture of the rest $K-1$ normal behaviour classes. In particular, Eqn. (18) is re-written as

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

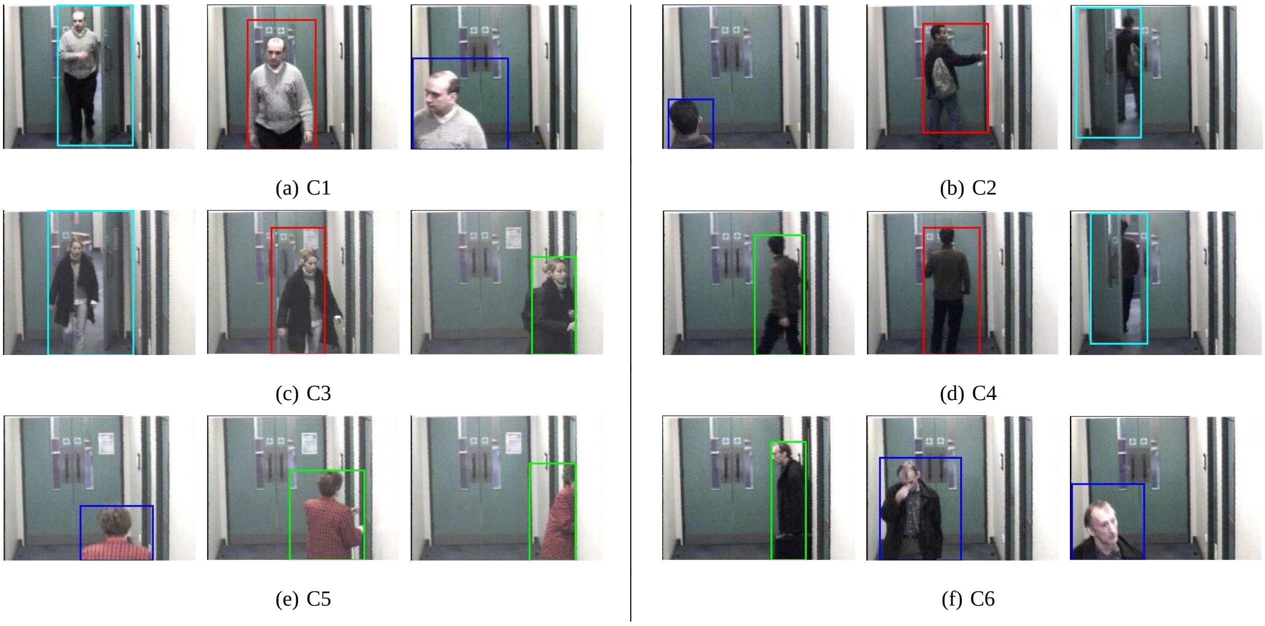
Note that r_k is a function of t and computed over time. $\mathbf{P}_t$ is reliably recognised as the kth behaviour class only when $1 \ll Th_r < r_k$. In our experiments we found that $Th_r = 10$ led to satisfactory results. When there are more than one r_k greater than Th_r, the behaviour pattern is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

For comparison, the commonly used *Maximum Likelihood* (ML) method recognises $\mathbf{P}_t$ as the kth behaviour 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 (e.g. a behaviour pattern can be explained away equally well by multiple plausible behaviour at its early stage). Compared to the ML method, our online LRT method holds the decision on behaviour 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, 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 (i.e. the denominator of Eqn. (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 behaviour classes give equally plausible explanations for a temporally incomplete behaviour 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 behaviour classes (i.e. $P(\mathbf{B}_k) = 1/K$ with $1 \leq k \leq K$), the MAP values for multiple classes can be high which will lead to 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 behaviour recognition task.

VI. EXPERIMENTS

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

A. Corridor Entrance/Exit Human behaviour Monitoring



(a) C1

(b) C2

(c) C3

(d) C4

(e) C5

(f) C6

Fig. 5. Behaviour patterns in a corridor entrance/exit scene. (a)–(f) show image frames of typical behaviour patterns belonging to the 6 behaviour classes listed in Table I. 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 colour scheme will be used for illustrating detected events in Fig. 8 and 9.

Dataset and feature extraction — A CCTV camera was mounted on the ceiling of an office entrance/exit corridor, monitoring people entering and leaving an office area (see Fig. 5). The office area is secured by an entrance-door which can only be opened by scanning an entry card on the wall next to the door (see middle frame in row (b) of Fig. 5). 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 behaviour occurring in the scene would be people entering or leaving either the office area or the side-doors, and walking towards the camera. Each behaviour pattern would normally last a few seconds. For this experiment, a dataset was collected over 5 different days consisting of 6 hours of video totalling 432000 frames captured at 20Hz with 320×240 pixels per frame. This dataset was then segmented into sections separated by any motionless intervals lasting for more than 30 frames. This resulted in 142 video segments of actual behaviour pattern instances. Each segment has on average 121 frames with the shortest 42 and longest 394.

Model training — A training set consisting of 80 video segments was randomly selected from the overall 142 segments without any behaviour class labelling of the video segments. The remaining 62 segments were used for testing the trained model later. This model training

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

TABLE I

THE 6 CLASSES OF BEHAVIOUR PATTERNS THAT MOST COMMONLY OCCURRED IN A CORRIDOR ENTRANCE/EXIT SCENE.

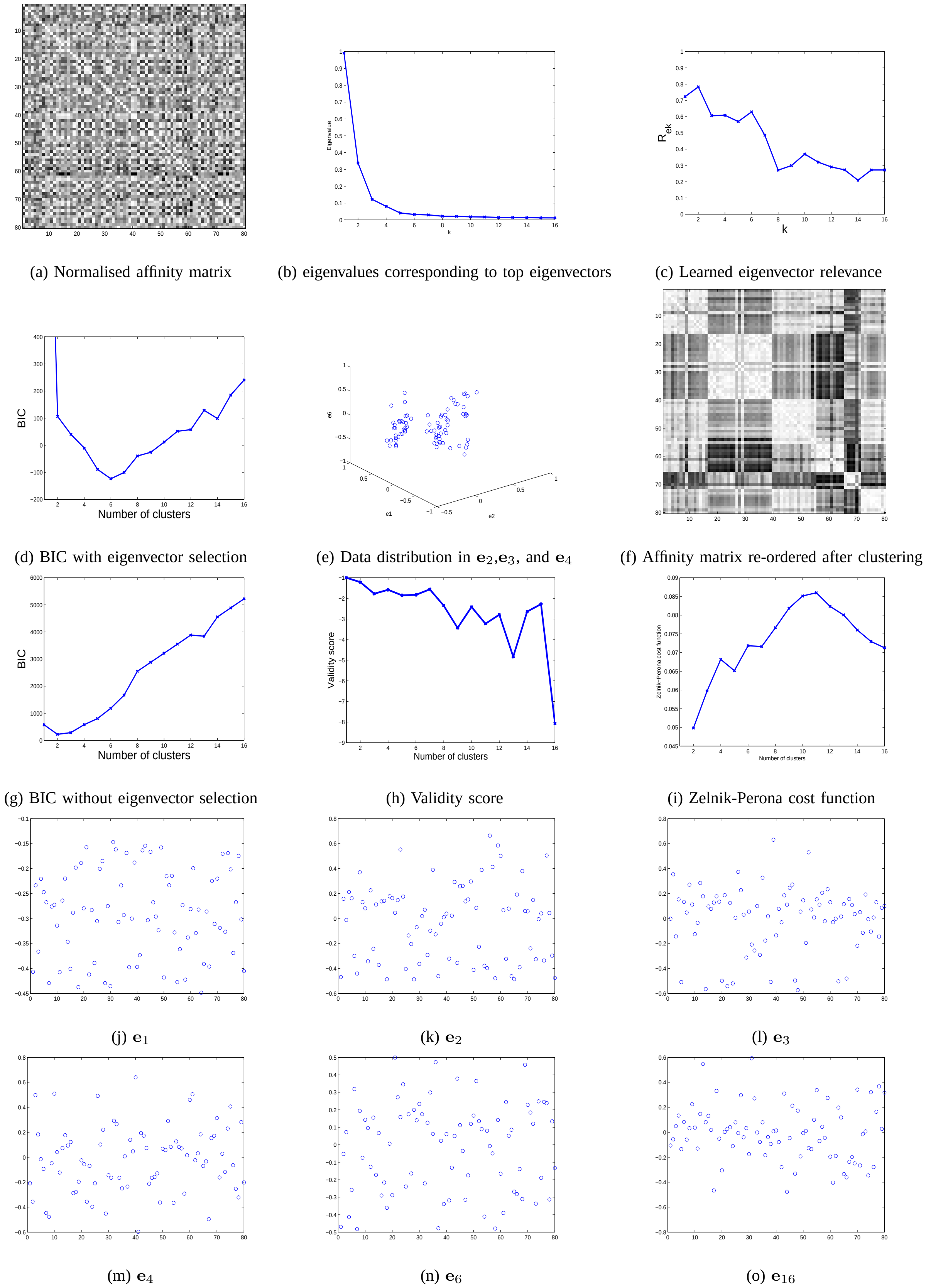
exercise was repeated 20 times and in each trial a different model was trained using a different random training set. This is in order to avoid any bias in the anomaly detection and normal behaviour recognition results to be discussed later. For comparative evaluation, alternative models were also trained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

Event detection for behaviour 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 behaviour 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 colour-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 (e.g. 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. 5(b) for an example). Nevertheless, the experiments presented below demonstrate that these 4 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 mis-identified based on local information (space and time) alone, resulting in large error margin in event detection. The fact that these events are also common constituents to different behaviour patterns reinforces the assumption that local events treated in isolation hold little discriminative information for behaviour profiling.

Model training using unlabelled data: Over the 20 trials, on average 6 eigenvectors were automatically determined as being relevant for clustering with the smallest 4 and largest 9. It is noted that all the selected eigenvectors were among the 10 largest eigenvectors of the normalised affinity matrices. The number of clusters for each training set was determined automatically as 6 in every trial. By observation, each discovered data cluster mainly contained samples corresponding to one of the 6 behaviour classes listed in Table I. 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 underestimated the class number. Fig. 6 shows an example of behaviour pattern clustering using unlabelled training sets. Note that compared to the synthetic data (see Fig. 4), the data we have for behaviour 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 Fig. 6 (j)-(o)). However, using only the relevant/informative eigenvectors, our algorithm can still discover the behaviour classes correctly. For each unlabelled training set, a normal behaviour model was constructed as a mixture of 6 MOHMMs as described in Section IV-E.

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

Anomaly detection — The behaviour models built using both labelled and unlabelled behaviour patterns were used to perform online anomaly detection. To measure the performance of the learned models on anomaly detection, each behaviour pattern in the testing sets was manually labelled as normal if there were similar patterns in the corresponding training sets and abnormal otherwise. A testing pattern was detected as being abnormal when Eqn. (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 anomaly detection



(a) Normalised 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 re-ordered after clustering

(g) BIC without eigenvector selection

(h) Validity score

(i) Zelnik-Perona cost function

(j) $\mathbf{e}_1$

(k) $\mathbf{e}_2$

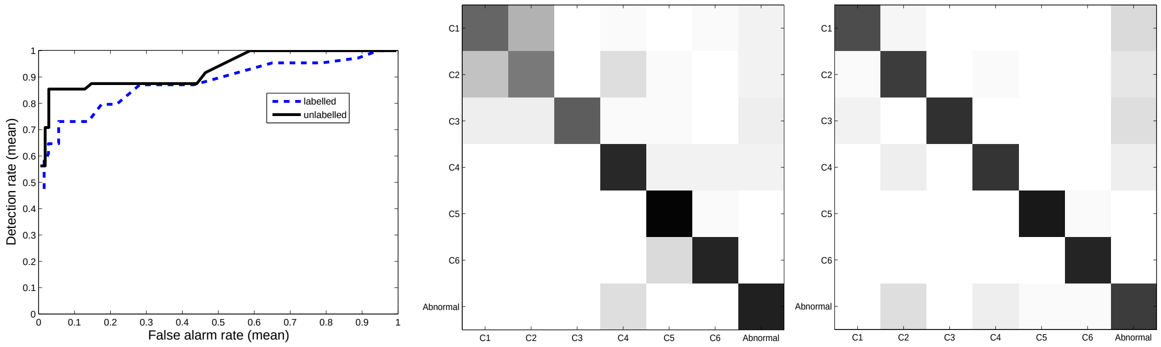
(l) $\mathbf{e}_3$

(m) $\mathbf{e}_4$

(n) $\mathbf{e}_6$

(o) $\mathbf{e}_{16}$

Fig. 6. An example of model training. (c) shows that the top 6 largest eigenvectors were determined as relevant features for clustering. (d) and (g) show the number of behaviour 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) ROC curves, labelled and unlabelled models (b) Confusion matrix, unlabelled models (c) Confusion matrix, labelled models

Fig. 7. (a): Comparing the performance of anomaly detection using corridor entrance/exit behaviour models trained by labelled and unlabelled data. The mean ROC curves were obtained over 20 trials. (b)&(c): Comparing the performance of behaviour recognition using models trained by labelled and unlabelled 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.

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

TABLE II

THE MEAN AND STANDARD DEVIATION OF THE ANOMALY DETECTION RATE AND FALSE ALARM RATES FOR CORRIDOR ENTRANCE/EXIT BEHAVIOUR MODELS TRAINED USING UNLABELLED AND LABELLED DATA. THE RESULTS WERE OBTAINED OVER 20 TRIALS WITH $Th_A = -0.2$.

rate and false alarm rate[4], which are defined as:

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

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 Eqn. (17)). Fig. 7(a) shows that the models trained using unlabelled data clearly outperformed those trained using labelled data. In particular, it is found that given the same Th_A the models trained using unlabelled datasets achieved higher anomaly detection rate and lower false alarm rate compared to those trained using labelled datasets (see also Table II and the last columns of the confusion matrices shown in Fig. 7(b)&(c)). Fig. 8 shows examples

[4] 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 true negative rate and false negative rate. Since we have 'true negative rate'=1-'false alarm rate' and 'false negative rate'=1-'anomaly detection rate', showing only anomaly detection rate and false alarm rate is adequate.

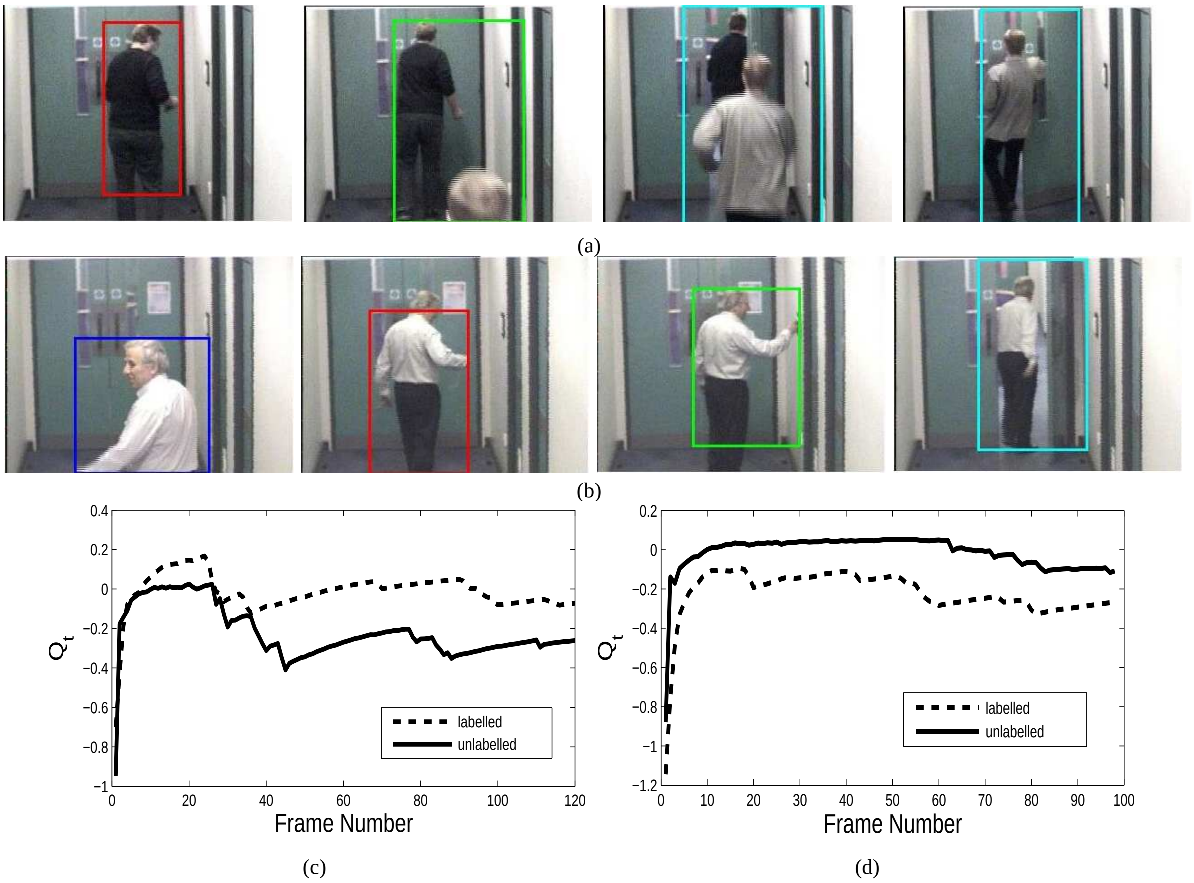


Fig. 8. Examples of anomaly detection in the corridor entrance/exit scene. (a)&(c): An abnormal behaviour pattern was detected as being abnormal by the model trained using an unlabelled dataset, while it was detected as being normal by the model trained using the same but labelled dataset. It shows a person sneaking into the office area without using an entry card. (b)&(d): A normal behaviour pattern which was detected correctly by the model trained using an unlabelled dataset, but detected as being abnormal by the model trained using the same but labelled dataset. 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 behaviour pattern to be abnormal. Th_A was set to -0.2.

of false alarm and mis-detection by models trained using labelled data. It is noted that the lower tolerance towards event detection errors was the main reason for the higher false alarm rate of models trained using labelled data (see Fig. 8(b)&(d) for an example).

Recognition of normal behaviour — To measure the recognition rate, the normal behaviour patterns in the testing sets were manually labelled into different behaviour classes. A normal behaviour pattern was recognised correctly if it was detected as normal and classified into a behaviour class containing similar behaviour patterns in the corresponding training set by the learned behaviour model. We first compare the performance of models trained using labelled and unlabelled data. Table III shows that the models trained using labelled data achieved slightly higher recognition rates compared to those trained using unlabelled data.

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

TABLE III

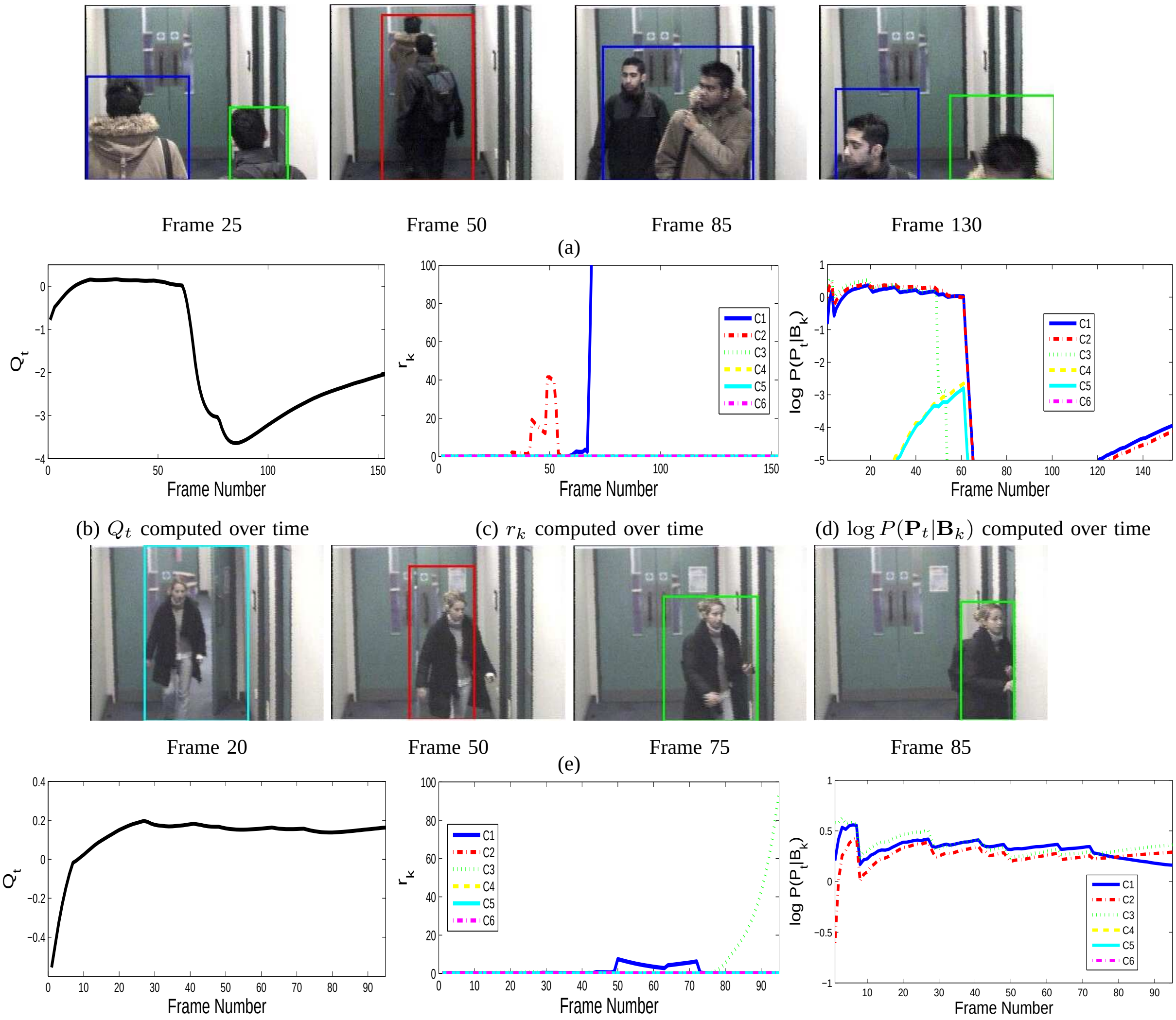
COMPARING THE PERFORMANCE OF MODELS TRAINED USING UNLABELLED AND LABELLED DATA ON CORRIDOR NORMAL BEHAVIOUR RECOGNITION. THE RESULTS WERE OBTAINED WITH $Th_A = -0.2$.

To have a more complete picture of the performance on normal behaviour recognition, the standard confusion matrix is utilised. Fig. 7(b) shows that when a normal behaviour pattern was not recognised correctly by a model trained using unlabelled data, it was most likely to be recognised as belonging to another normal behaviour class. On the other hand, Fig. 7(c) shows that for a model trained by labelled data, a normal behaviour pattern was most likely to be wrongly detected as an anomaly if it was not recognised correctly. This contributed to the higher false alarm rate for the model trained by labelled data.

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

B. Aircraft Docking Area Behaviour Monitoring

Dataset 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 behaviour 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 2Hz which is common for CCTV surveillance videos. Each image frame has a size of 320×240 pixels. Our database for the experiments consists of 72776 frames of video data (around 10 hours of recording) that cover different times of different days under changing lighting conditions, from early morning, midday to late afternoon. The video was segmented automatically using an online segmentation algorithm proposed in [27], giving 59 video segments of actual behaviour pattern instances. Each segment has on average 428



Frame 25 Frame 50 Frame 85 Frame 130

(a)

(b) Q_t computed over time (c) r_k computed over time (d) log $P(\mathbf{P}_t|\mathbf{B}_k)$ computed over time

Frame 20 Frame 50 Frame 75 Frame 85

(e)

(f) Q_t computed over time (g) r_k computed over time (h) log $P(\mathbf{P}_t|\mathbf{B}_k)$ computed over time

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

frames with the shortest 74 and longest 2841.

Model training — A training set now consisted of 40 video segments and the remaining 19 were used for testing. As in the corridor behaviour monitoring experiments, 20 trials were conducted, each of which had a different random training set. To compare models trained using unlabelled data with those trained using labelled data, in each training session a model

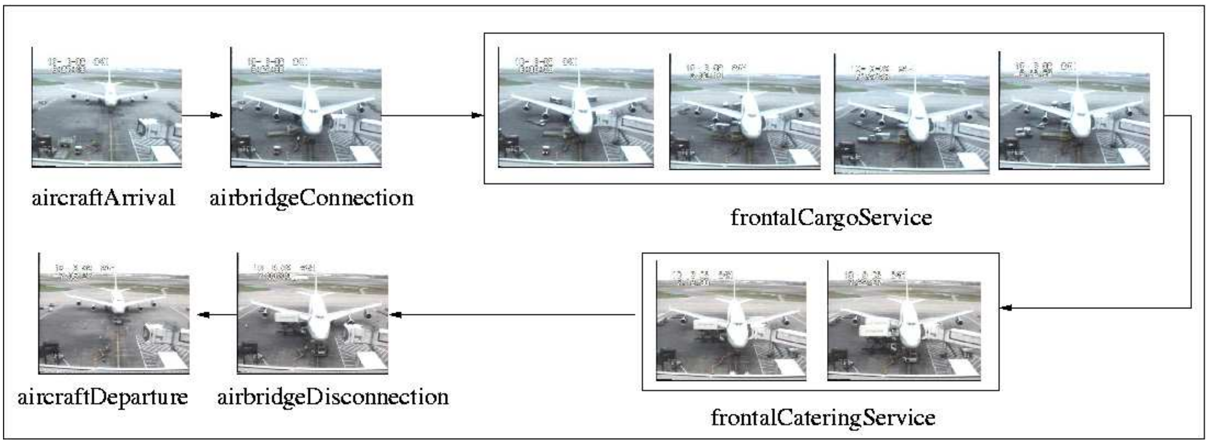


Fig. 10. Typical, visually detectable behaviour patterns in an aircraft docking scene.

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

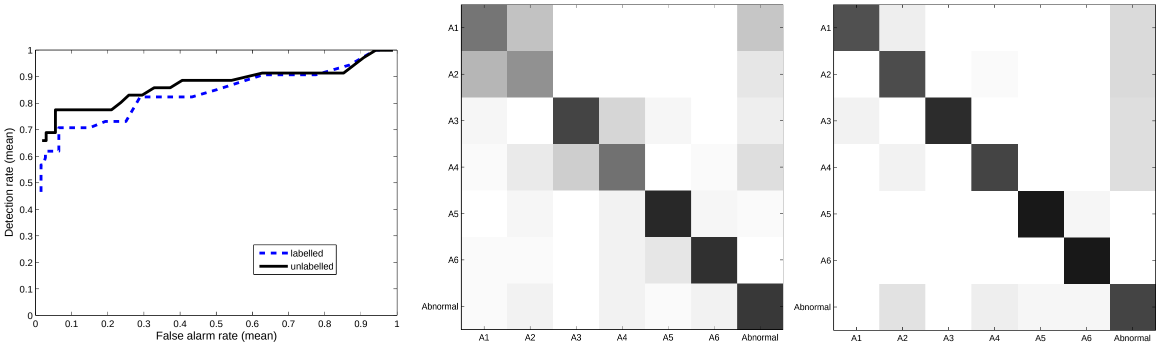
TABLE IV

SIX CLASSES OF COMMONLY OCCURRED BEHAVIOUR PATTERNS IN THE AIRPORT SCENE.

was also trained using an identical but labelled training set. On average 9 behaviour classes were manually identified for the labelled training sets in each trial. Six classes were always identified in each training set (see Table IV). On average they accounted for 74% of the labelled training data.

Given a training set, discrete events were detected and classified, resulting in 8 classes of events corresponding to the common constituents of all behaviour 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 unlabelled training behaviour patterns were represented based on the event detection results and clustered to build a composite behaviour model. On average 7 eigenvectors were automatically determined as being relevant for clustering with the smallest 4 and largest 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 6 discovered data clusters mainly contained samples corresponding to one of the 6 behaviour classes listed in Table IV. In each session, a model was also built using labelled data for comparison.



(a) ROC curves, labelled and unlabelled models (b) Confusion matrix, unlabelled models (c) Confusion matrix, labelled models

Fig. 11. (a): Comparing the performance of anomaly detection using aircraft docking behaviour models trained by labelled and unlabelled data. The mean ROC curves were obtained over 20 trials. (b)&(c): Comparing the performance of behaviour recognition using models trained by labelled and unlabelled 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.

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

TABLE V

The mean and standard deviation of the anomaly detection rate and false alarm rates for aircraft docking behaviour models trained using unlabelled and labelled data. The results were obtained over 20 trials with $Th_A = -0.5$.

Anomaly detection — Each behaviour pattern in a testing set was detected as being abnormal when Eqn. (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 behaviour monitoring experiments). Fig. 11(a) shows that the models trained using unlabelled data outperformed those trained using labelled data. The results are slightly inferior to those obtained in the corridor entrance/exit behaviour modelling experiments (comparing Fig. 11(a) with Fig. 7(a)). It is not surprising due to the factor that the data collected in this outdoor scenario are much more noisy and also more sparse while the behaviour captured in the data are more complicated. It is not surprising since the data collected in this outdoor scenario are much more noisy and sparse, and the behaviour captured in the data are more complicated. Fig. 12(b)&(f) show examples of online reliable anomaly detection using our run-time anomaly measure.

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

TABLE VI

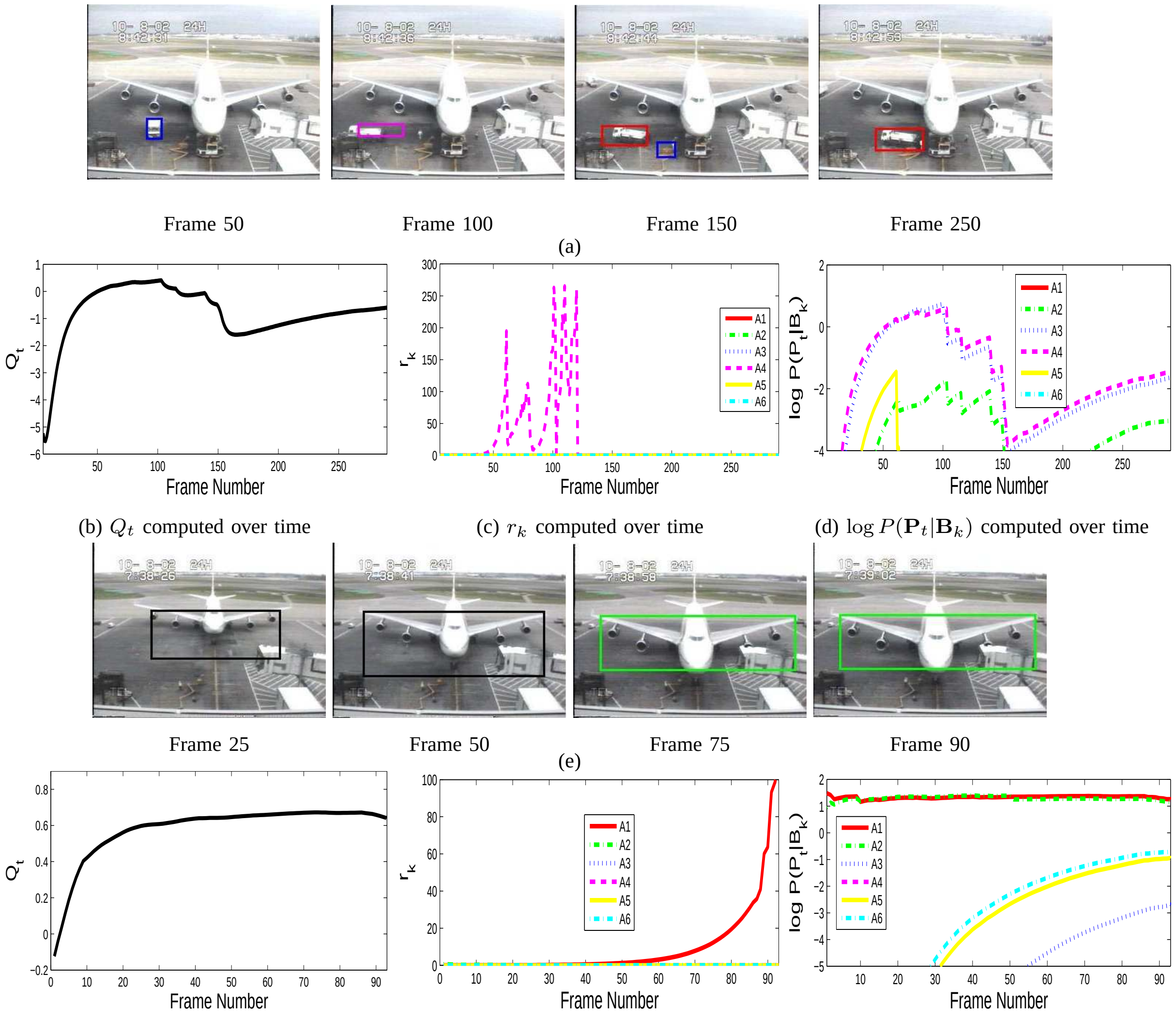
COMPARING THE PERFORMANCE OF MODELS TRAINED USING UNLABELLED AND LABELLED DATA ON AIRCRAFT DOCKING NORMAL BEHAVIOUR RECOGNITION. THE RESULTS WERE OBTAINED WITH $Th_A = -0.5$.

Recognition of normal behaviour patterns — The normal behaviour recognition results obtained using models trained by unlabelled and labelled data are illustrated in Table VI and Fig. 11(b)&(c). 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 unlabelled data. However, when not being recognised correctly, a normal behaviour pattern is more likely to be detected as anomaly using a labelled data trained model. This resulted in higher false alarm rate. Examples of online reliable behaviour 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 behaviour patterns can be reliably and promptly recognised after sufficient visual evidence had become available to overcome the ambiguities among different behaviour classes.

VII. Discussions and Conclusions

The key findings of our experiments are summarised and discussed as below:

1) Our experiments show that a behaviour model trained using an unlabelled dataset is superior to a model trained using the same but labelled dataset in detecting anomaly from an unseen video. The former model also outperforms the latter in distinguishing abnormal behaviour patterns from normal ones contaminated by errors in behaviour representation. In comparison, a model trained using manually labelled data may have an advantage in explaining data that are well-defined. However, training using labelled data does not necessarily help a model with identifying novel instances of abnormal behaviour patterns as the model tends to be brittle and less robust in dealing with instances of behaviour that are not clear-cut in an open-world scenario (i.e. the number of expected normal and abnormal behaviour cannot be pre-defined exhaustively).



Frame 50 Frame 100 Frame 150 Frame 250

(a)

(b) Q_t computed over time (c) r_k computed over time (d) log $P(\mathbf{P}_t|\mathbf{B}_k)$ computed over time

Frame 25 Frame 50 Frame 75 Frame 90

(e)

(f) Q_t computed over time (g) r_k computed over time (h) log $P(\mathbf{P}_t|\mathbf{B}_k)$ computed over time

Fig. 12. Compare our LRT method with ML method for online normal behaviour recognition in an aircraft docking scene. (a): An abnormal behaviour 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 anomaly from Frame 147 till the end based on Q_t. (c): The behaviour pattern between Frame 53 to 145 was recognised reliably as A4 using LRT before becoming abnormal and being detected using Q_t. (d) The behaviour pattern was wrongly recognised as A3 between Frame 80 to 98 using the ML method. (e): A normal A1 behaviour pattern. (f): The behaviour pattern was detected as normal throughout based on Q_t. (g): It was recognised reliably as A1 from Frame 73 till the end using LRT. (h): It was wrongly recognised as A2 between Frame 12 to 49 using ML. In (a) and (e), detected events are illustrated using the same colour scheme as in Fig. 1.

2) Our eigenvector selection based spectral clustering algorithm is capable of discovering natural grouping of behaviour patterns in the training data. Our experiments show that our simple data-driven eigenvector selection algorithm works well on real world behaviour data. Furthermore, the eigenvector selection step of the algorithm is critical for determining the optimal number of behaviour pattern classes.

3) Our online Likelihood Ratio Test (LRT) based normal behaviour recognition method is

superior to the conventional ML based method. Since normal behaviour recognition is performed after anomaly detection, it is plausible to construct the alternative model as a mixture of other normal behaviour patterns. This is the key to make a LRT method work because the performance of a LRT method depends on the accuracy of constructing the alternative model.

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 (e.g. 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 [1] like EM algorithm to the mixture of DBNs (Eqn. (14)) to learn the behaviour 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 behaviour classes automatically, which corresponds to the number of mixtures in the directly learned model. However, in practice learning the model directly brings about the initialisation problem which could lead to poorer results compared to the proposed stepped approach. The initialisation problem always exists for a model based clustering algorithm (e.g. Gaussian Mixture Models). However, the problem becomes rather acute in the case of clustering using 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 initialisation 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 behaviour profiling framework is flexible in the sense that it allows for the use of different behaviour segmentation, representation, and affinity measurement as long as a behaviour 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 behaviour recognition and anomaly detection. The framework is fully unsupervised and consisted of a number of key components, namely a discrete event based behaviour representation, a DBN based behaviour affinity measure, a spectral clustering algorithm with feature and model selection, a composite behaviour model based on a mixture of DBNs, a run-time accumulative anomaly measure, and an online LRT based normal behaviour recognition method. The effectiveness and robustness of our approach is demonstrated through experiments using datasets 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 towards a general-purpose anomaly detection method which 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 analysis of flow dynamics for extremely crowded dynamic scenes and investigating the possibility of combining rule-based behaviour profiling approaches with statistical learning based approaches. Another drawback of the work is that it does not cope with changes of visual context which 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 behaviour 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 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 generalised to others? Again, human has the ability to generalise 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 re-do 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. *Ann. Math. Stat.*, 37:1554–1563, 1966.

[2] O. Boiman and M. Irani. Detecting irregularities in images and in video. In *IEEE International Conference on Computer Vision*, pages 462–469, 2005.

[3] H. Dee and D. Hogg. Detecting inexplicable behaviour. In *British Machine Vision Conference*, pages 477–486, 2004.

[4] A. Dempster, N. Laird, and D. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[5] T. Duong, H. Bui, D. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-Markov model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 838–845, 2005.

[6] J. Dy, C. Brodley, A. Kak, L. Broderick, and A. Aisen. Unsupervised feature selection applied to content-based retrival of lung images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 373–378, 2003.

[7] Z. Ghahramani. Learning dynamic bayesian networks. In *Adaptive Processing of Sequences and Data Structures. Lecture Notes in AI*, pages 168–197, 1998.

[8] S. Gong and T. Xiang. Recognition of group activities using dynamic probabilistic networks. In *IEEE International Conference on Computer Vision*, pages 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. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1031–1038, 2005.

[10] A. Higgins, L. Bahler, and J. Porter. Speaker verification using randomized phrase prompting. *Digital Signal Processing*, pages 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 Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1154–1166, 2004.

[13] L. R.Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[14] R. J. Morris and D. C. Hogg. Statistical models of object interaction. *International Journal of Computer Vision*, 37(2):209–215, 2000.

[15] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *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 Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, August 2000.

[17] A. Panuccio, M. Bicego, and V. Murino. A hidden Markov model-based approach to sequential data clustering. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 734–742, London, UK, 2002. Springer-Verlag.

[18] F. Porikli. Trajectory distance metric using hidden Markov model based representation. In *The Sixth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2002.

[19] F. Porikli and T. Haga. Event detection by eigenvector decomposition using object and frame features. In *CVPRW*, pages 114–121, 2004.

[20] S. Roberts, D. Husmeier, I. Rezek, and W. Penny. Bayesian approaches to Gaussian mixture modelling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1133–1142, 1998.

[21] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[22] Y. Shan, H. S. Sawhney, and A. Pope. Measuring the similarity of two image sequence. In *Asian Conference on Computer Vision*, 2004.

[23] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[24] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–758, August 2000.

[25] Y. Weiss. Segmentation using eigenvectors: a unifying view. In *IEEE International Conference on Computer Vision*, pages 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. Acoustic, Speech and Signal Processing*, pages 1870–1878, 1990.

[27] T. Xiang and S. Gong. Activity based video content trajectory representation and segmentation. In *British Machine Vision Conference*, pages 177–186, 2004.

[28] T. Xiang and S. Gong. Video behaviour profiling and abnormality detection without manual labelling. In *IEEE International Conference on Computer Vision*, pages 1238–1245, 2005.

[29] T. Xiang, S. Gong, and D. Parkinson. Autonomous visual events detection and classification without explicit object-centred segmentation and tracking. In *British Machine Vision Conference*, pages 233–242, 2002.

[30] S. Yu and J. Shi. Multiclass spectral clustering. In *IEEE International Conference on Computer Vision*, pages 313–319, 2003.

[31] L. Zelnik-Manor and M. Irani. Event based video analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[32] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *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. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 611–618, 2005.

[34] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826, 2004.