Incremental Activity Modeling in Multiple Disjoint Cameras

Chen Change Loy, *Member*, *IEEE*, Tao Xiang, and Shaogang Gong

Abstract—Activity modeling and unusual event detection in a network of cameras is challenging, particularly when the camera views are not overlapped. We show that it is possible to detect unusual events in multiple disjoint cameras as context-incoherent patterns through incremental learning of time delayed dependencies between distributed local activities observed within and across camera views. Specifically, we model multicamera activities using a Time Delayed Probabilistic Graphical Model (TD-PGM) with different nodes representing activities in different decomposed regions from different views and the directed links between nodes encoding their time delayed dependencies. To deal with visual context changes, we formulate a novel incremental learning method for modeling time delayed dependencies that change over time. We validate the effectiveness of the proposed approach using a synthetic data set and videos captured from a camera network installed at a busy underground station.

Index Terms—Unusual event detection, multicamera activity modeling, time delay estimation, incremental structure learning.

1 Introduction

Wide area and complex public scenes are often monitored by multiple cameras, the majority of which have disjoint views. In this study, we address the problem of detecting and localizing unusual events[1] occurring in a crowded public scene monitored by multiple disjoint cameras with nonoverlapping field of views (FOV). In particular, our method aims to detect *global unusual events*, i.e., context-incoherent patterns that span across multiple disjoint camera views.

To solve the problem, global modeling of activity patterns is indispensable because unusual events can take place globally across multiple disjoint cameras and often appear normal in isolated camera views. An individual inspection on each view would fail to detect such an unusual event since a global behavior interpretation is not achievable based solely on visual evidences captured locally within a single view.

Global activity modeling and unusual event detection across multiple disjoint cameras in a crowded public scene is intrinsically difficult due to several inextricable factors:

1. **Unknown and arbitrary intercamera gaps**—Unknown and often large separation of cameras in space causes temporal discontinuity in visual observations, i.e., a global activity can only be observed partially in different views while portions of the activity sequence may be unobserved due to the intercamera gaps. To further complicate the matter, two widely separated camera views may include an arbitrary number of entry/exit locations in the gap, where existing objects can disappear and new objects can appear, causing uncertainty in understanding and correlating activities in both camera views.
2. **Intercamera visual variations**—Objects moving across camera views often experience drastic variations in their visual appearances due to different illumination conditions, camera orientations, and changes in object pose.
3. **Low-quality videos captured in crowded scene**—In a typical public scene, the sheer number of objects causes severe and continuous interobject occlusions. Applying conventional object-centered strategy that requires explicit object segmentation and tracking within/across camera views can be challenging. Tracking can be further compounded by the typically low temporal and spatial resolutions of surveillance video, where large spatial displacement is observed in moving objects between consecutive frames.
4. **Visual context variations**—In an unconstrained environment, visual context changes are ineluctable and may occur gradually or abruptly. In particular, gradual context change may involve gradual behavior drift over time, e.g., different volumes of crowd flow at different time periods. On the other hand, abrupt context change implicates more drastic changes such as camera angle adjustment, removal/addition of camera from/to a camera network. Both gradual and abrupt changes cause transitions and modifications of intercamera activity dependency over time.

Due to the aforementioned challenges, visual observations from different camera views are inevitably noisy and partial, making the meaning of activity ambiguous.

To mitigate the effects of the aforementioned *factors 1 and 2*, we believe the key is to learn a *global visual context* to associate partial observations of activities observed across camera views. Activities in a public space are inherently

[1] Rare or abnormal events that should be reported for further examination.

• *The authors are with the School of Electrical Engineering and Computer Science, Queen Mary University of London, London E1 4NS, United Kingdom. E-mail: {ccloy, txiang, sgg}@eecs.qmul.ac.uk.*

Manuscript received 3 Dec. 2010; revised 17 June 2011; accepted 29 Oct. 2011; published online 13 Dec. 2011.
Recommended for acceptance by G. Mori.
For information on obtaining reprints of this article, please send E-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2010-12-0921.
Digital Object Identifier no. 10.1109/TPAMI.2011.246.

context-aware, often exhibited through constraints imposed by scene layout and the correlated activities of other objects, both in the same camera view and other views. The global visual context should encompass spatial and temporal context, defining where and when a partial observation occurs, as well as a correlation context specifying the expectation inferred from the correlated behaviors of other objects in the camera network, i.e., the dependency and associated time delay between activities. To this end, we propose to model global visual context by learning global dependencies and the associated time delays between distributed local activities. Specifically, we formulate a novel Time Delayed Probabilistic Graphical Model (TD-PGM), whose nodes represent activities in different decomposed regions from different views and the directed links between nodes encoding the time delayed dependencies between the activities. Consequently, global unusual events can be detected and localized as *context-incoherent* patterns through inspecting the consistency between node observation and graph propagation in the learned model.

To circumvent the problem caused by *factor 3*, our model employs a holistic activity representation rather than conventional trajectory-based representation that relies on explicit object-centered segmentation and tracking (see Section 3.1). Therefore, it can be applied to low-quality public scene surveillance videos featuring severe interobject occlusions for robust multicamera unusual event detection.

To cope with the visual context variations (*factor 4*), we treat the dependency learning problem as an incremental graph structure learning (i.e., to discover conditional dependency links between a set of nodes) and parameter learning task (i.e., to learn the parameter associated with the links). In particular, we formulate a novel incremental two-stage structure learning approach to learn the updated camera network structure in accordance to the current visual context, without any prior knowledge and assumptions on the camera topology.

Extensive evaluations are conducted on a synthetic data set and 167 hours of videos acquired from a camera network installed at a busy underground station.

2 Related Work

There has been a considerable amount of work for activity understanding and unusual event detection in surveillance videos, but mostly devoted to a single camera scenario [1], [2], [3], [4]. There also exist incremental learning methods [2], [5] that accommodate visual context changes over time. These methods are not directly applicable to scenarios involving multiple disjoint cameras since there is no mechanism to discover and quantify arbitrary time delays among activities observed across nonoverlapping views.

Recently, a number of methods have been proposed to model activity and detect unusual event across multiple disjoint cameras. One of the popular approaches is to reconstruct the global path taken by an object by merging its trajectories observed in different views, followed by a standard single-view trajectory analysis approach [6]. With this approach, one must address the camera topology inference problem [7], [8] and the trajectory correspondence problem [9], both of which are far from being solved. Wang et al. [10] propose an alternative trajectory-based method that bypasses the topology inference and correspondence problems. However, the method cannot cope with busy scenes and it is limited to capturing only co-occurrence relationships among activity patterns, but not the time delayed dependencies between local activities cross views.

Zhou and Kimber [11] attempt to circumvent unreliable tracking by using an event-based representation together with a Coupled Hidden Markov Model (CHMM) for activity modeling. However, the model is not scalable to a large camera network and the CHMM chain's connectivity has to be manually defined to reflect neighboring relationships of cameras. Moreover, the model is restricted to capturing first-order temporal dependency, which is not suitable for modeling cross-camera activity dependencies with arbitrary time delays.

A closely related work is the modeling of transition time distribution between entry/exit events in two cameras [7], [8]. However, these methods rely on intracamera tracking to detect entry/events in order to infer the transition time statistics. As mentioned in Section 1, explicit object segmentation and tracking are nontrivial in a crowded scene, especially given video captured in low temporal and spatial resolutions. Importantly, these methods do not address the unusual event detection problem.

Our approach is centered around a novel incremental two-stage structure learning algorithm for a TD-PGM. There is a rich literature on graphical model structure learning. Previous methods can be categorized into either constraint-based methods [12], [13, or scored-searching-based methods [14], [15]. Hybrid approaches have also been proposed to combine both methods above in order to improve computational efficiency and prediction accuracy in structure learning [16], [17]. Existing hybrid approaches, however, are not capable of learning graph dependencies among multiple time series with unknown time delays. To overcome this problem, we propose a new hybrid approach that combines a scored-searching-based method with an information theoretic-based analysis for time delay estimation.

Several approaches for an incremental structure learning of probabilistic graphical models have been proposed in the past [18], [19]. A notable method is presented by Friedman and Goldszmidt [18], whereby a structure is updated sequentially without having to store all earlier observations. Our incremental structure learning method is similar in spirit to that in [18], but with several key differences that make our approach more suitable for incremental activity modeling in a large distributed camera network, for which tractability and scalability are more critical. First, our approach allows more tractable structure update for a large camera network. Second, the proposed method requires less memory. Specifically, the prior work [18] employs a single-stage greedy hill-climbing (GHC) structure learning [20] without any constraint on structure search space. The method is thus intractable given a large graph with hundred of nodes [21]. It also stores a large amount of sufficient statistics to update the dependency links given a large graph structure. In contrast, our two-stage structure learning approach achieves a more tractable learning by exploiting the time delay information to derive an ordering

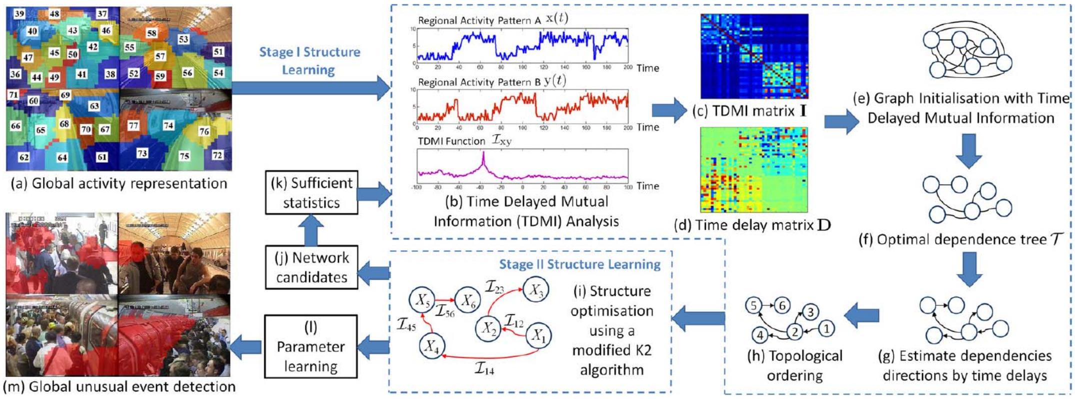

39
48
37
40
43
46
58
53
47
45
50
42
55
57
51
36
44
49
41
38
52
59
56
54
71
60
69
63
66
65
68
70
67
77
74
76
62
64
61
73
75
72
(a) Global activity representation
Stage I Structure Learning
Regional Activity Pattern A $x(t)$
Regional Activity Pattern B $y(t)$
TDMI Function $\mathcal{I}_{xy}$
Time
(b) Time Delayed Mutual Information (TDMI) Analysis
(c) TDMI matrix $\mathbf{I}$
(d) Time delay matrix $\mathbf{D}$
(e) Graph Initialisation with Time Delayed Mutual Information
(f) Optimal dependence tree $\mathcal{T}$
(g) Estimate dependencies directions by time delays
(h) Topological ordering
Stage II Structure Learning
X_5
X_6
X_3
X_2
X_1
X_4
$\mathcal{I}_{23}$
$\mathcal{I}_{56}$
$\mathcal{I}_{12}$
$\mathcal{I}_{45}$
$\mathcal{I}_{14}$
(i) Structure optimisation using a modified K2 algorithm
(k) Sufficient statistics
(j) Network candidates
(l) Parameter learning
(m) Global unusual event detection

Fig. 1. A diagram illustrating our approach for incremental learning of global time delayed dependencies between activities observed in multiple disjoint cameras.

constraint for reducing the search space, as well as for pruning less probable candidate structures and the associated sufficient statistics during the searching process, therefore resulting in lower memory consumption.

The main contributions of this work are:

1. To the best of our knowledge, this work is the first study on modeling time delayed activity dependencies for real-time detection of global unusual events across distributed multicamera views of busy public scenes.
2. Existing studies [10], [11], [22] generally assume an activity model that remain static once learned; the problem of incremental global activity modeling in multiple disjoint cameras has not been addressed before. To cope with the inevitable visual context changes over time, a novel incremental two-stage structure learning method is proposed to discover and quantify an optimized time delayed dependency structure globally.

3 Global Activity Dependency Modeling

3.1 Global Activity Representation

An overview of the key steps of our approach is given in Fig. 1. To facilitate global activity understanding across nonoverlapping camera views, it is necessary to decompose each camera view into regions (Fig. 1a) where different activity patterns are observed (e.g., decompose a traffic junction into different lanes and waiting zones). We refer to an activity that takes place locally in a region (e.g., driving in a lane or parking in a waiting zone) as a *regional activity*. To this end, we first perform foreground extraction and separate the foreground pixels into static and moving activity patterns. Specifically, given a set of detected foreground pixels, the moving foreground pixels are identified by performing frame differencing between successive frames. Foreground pixels that do not belong to moving foreground pixels are then classified as static foreground pixels. Next, the approach proposed in [23] is adopted to cluster a scene using spectral clustering [24] based on correlation distances of local block spatiotemporal activity patterns. This results in n regions across all views, which are indexed in a common reference space.

Given the decomposed scene, activity patterns observed over time in the ith region are represented as a bivariate time series: $\hat{\mathbf{u}}_i = (\hat{u}_{i,1}, \ldots, \hat{u}_{i,t}, \ldots)$ and $\hat{\mathbf{v}}_i = (\hat{v}_{i,1}, \ldots, \hat{v}_{i,t}, \ldots)$, where $\hat{u}_{i,t}$ represents the percentage of static foreground pixels within the ith region at time t, while $\hat{v}_{i,t}$ is the percentage of pixels within the region that are classified as moving foreground.

To enable the proposed TD-PGM to model a scene with a fixed and finite number of states, we feed the two-dimensional time series $(\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i)$ of length T as an input to a Gaussian Mixture Model (GMM). The GMM is trained via Expectation-Maximization (EM) with the number of component K_i determined by automatic model order selection using Bayesian Information Criterion (BIC). The learned GMM is then used to classify activity patterns detected in each region at each frame into one of the K_i components. The typical value for K_i ranges from 5 to 10, depending on the complexity of the regional activity patterns in our experiments. Activity patterns in the ith region over time are thus represented using the class labels and denoted as a one-dimensional time series:

$$\mathrm{x}_i(t) = (x_{i,1}, \ldots, x_{i,t}, \ldots), \tag{1}$$

where $x_{i,t} \in \{1, 2, \ldots, K_i\}$ and $i = 1, \ldots, n$.

It is worth pointing out that the quality of scene decomposition or local behavior grouping will have an effect on the learned global activity model. Both under and oversegmentation of a scene will have an adverse effect. In particular, undersegmentation may not produce local scene regions that encompass distinctive set of activities, which will cause difficulties in learning the time delayed correlations between regions. In comparison, oversegmentation is more likely to produce distinctive partitioning of the local activities. However, one would expect higher computational cost in subsequent analyses given the oversegmented regions. Overall, oversegmentation is less an issue provided that the increase of model complexity would not render the model learning intractable. The scene segmentation method adopted [23] is flexible in using different types of representation. If higher frame rate video is available, motion information [25], [26], [27] can be readily used to improve the segmentation result.

3.2 Time Delayed Probabilistic Graphical Model

We model time delayed dependencies among regional activity patterns using a TD-PGM (Fig. 1i). A TD-PGM is defined as $B = \langle G, \Theta \rangle$, which consists of a directed acyclic graph (DAG) G whose nodes represent a set of discrete random variables $\mathbf{X} = \{X_i | i = 1, 2, \ldots, n\}$, where X_i is the ith variable representing activity patterns observed in the ith region. Specific value taken by a variable X_i is denoted as x_i. A stream of values x_i of variable X_i is denoted as $\mathrm{x}_i(t) = (x_{i,1}, \ldots, x_{i,t}, \ldots)$ (see (1)).

The model is quantified by a set of parameters denoted by Θ specifying the conditional probability distribution (CPD), $p(X_i|\mathbf{Pa}(X_i))$. Since all the observations in the model are finite-state variables due to the GMM clustering, the CPD between a child node X_i and its parents $\mathbf{Pa}(X_i)$ in G is represented using a multinomial probability distribution. Consequently, Θ contains a set of parameters $\boldsymbol{\theta}_{x_i|\mathbf{pa}(X_i)} = p(x_i|\mathbf{pa}(X_i))$ for each possible discrete value x_i of X_i and $\mathbf{pa}(X_i)$ of $\mathbf{Pa}(X_i)$. Here, $\mathbf{Pa}(X_i)$ represents the set of parents of X_i, and $\mathbf{pa}(X_i)$ is an instantiation of $\mathbf{Pa}(X_i)$.

Conditional independence is assumed. The dependencies among variables are represented through a set of directed edges $\mathbf{E}$, each of which points to a node from its parents on which the distribution is conditioned. Given any two variables X_i and X_j, a directed edge from X_i to X_j is denoted as $X_i \rightarrow X_j$, where $(X_i, X_j) \in \mathbf{E}$ and $(X_j, X_i) \notin \mathbf{E}$. Note that the $p(X_i|\mathbf{Pa}(X_i))$ are not quantified using a common time index but with relative time delays that are discovered using Time Delayed Mutual Information (TDMI) discussed in the next section.

Other notations we use are given as follows: The number of states of X_i is r_i, and the number of possible configurations of $\mathbf{Pa}(X_i)$ is q_i. A set of discrete value x_i across all variables is given as $\mathbf{x} = \{x_i | i = 1, 2, \ldots, n\}$. Consequently, a collection of m cases of $\mathbf{x}$ is denoted as $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$. The number of cases of $(x_i, \mathbf{pa}(X_i))$ in $\mathcal{X}$ is represented as $N_{x_i|\mathbf{pa}(X_i)}$, specifically, $N_{ijk} = N_{x_i=k|\mathbf{pa}(X_i)=j}$.

3.3 Two-Stage Structure Learning

The optimal structure of the TD-PGM, B encodes the time delayed dependencies that we aim to discover and quantify. In the following Sections 3.3.1 and 3.3.2, we will first explain our two-stage structure learning algorithm operating in a batch mode. This shall facilitate explanation of the incremental extension of this approach described in Section 3.6.

3.3.1 Constraint-Based Learning with Time Delayed Mutual Information Analysis

There are two stages in our structure learning method. In the first-stage, constraint-based learning (Figs. 1b, 1c, 1d, 1e, 1f, 1g, and 1h), we wish to obtain a prior graph structure which can be further used to derive an ordering constraint. The constraint is propagated to the second-stage scored-searching-based learning (Fig. 1i) to *reduce and constrain the structure search space*, by eliminating any candidate structure inconsistent with the constraint. This consequently leads to a significant computational speed up in the second-stage learning process. Let us now detail the steps involved in the first-stage learning:

- *Step 1—TDMI analysis* (Figs. 1b, 1c, and 1d))—Time Delayed Mutual Information [28] analysis is explored here to learn initial time delayed association between each pair of regional activity patterns. The TDMI was first introduced by Fraser and Swinney [28] for determining delay parameter in a chaotic dynamical system through measuring the Mutual Information (MI) between a time series $\mathrm{x}(t)$ and a time shifted copy of itself $\mathrm{x}(t+\tau)$ as a function of time delay τ. The main rationale behind the use of TDMI is that if two regional activity patterns are dependent, information conveyed by a region would provide a large amount of information on another region.

  In TDMI analysis, if one treats two arbitrary regional activity patterns as time series data and denotes them as $\mathrm{x}(t)$ and $\mathrm{y}(t)$, respectively, the TDMI of $\mathrm{x}(t)$ and time shifted $\mathrm{y}(t+\tau)$ can be written as follows:

$$\mathrm{I}(\mathrm{x}(t);\ \mathrm{y}(t+\tau)) = \sum_{j=1}^{K_\mathrm{x}} \sum_{k=1}^{K_\mathrm{y}} p_{\mathrm{xy}}(j,k) \log_2 \frac{p_{\mathrm{xy}}(j,k)}{p_\mathrm{x}(j) p_\mathrm{y}(k)}, \tag{2}$$

  where $p_\mathrm{x}(\cdot)$ and $p_\mathrm{y}(\cdot)$ denote the marginal probability distribution functions of $\mathrm{x}(t)$ and $\mathrm{y}(t+\tau)$, respectively, while $p_{\mathrm{xy}}(\cdot)$ is the joint probability distribution function of $\mathrm{x}(t)$ and $\mathrm{y}(t+\tau)$. The probability distribution functions are approximated by constructing histograms with K_i bins, each of which corresponds to one GMM class discovered using approach described in Section 3.1. Note that $\mathrm{I}(\mathrm{x}(t);\ \mathrm{y}(t+\tau)) \geq 0$ with the equality if and only if $\mathrm{x}(t)$ and $\mathrm{y}(t+\tau)$ are independent. If $\tau = 0$, TDMI is equivalent to MI of $\mathrm{x}(t)$ and $\mathrm{y}(t)$.

  Subsequently, a TDMI function $\mathcal{I}_{\mathrm{xy}}\ (\tau)$ is obtained as a sequence of TDMI values $\mathrm{I}(\mathrm{x}(t);\ \mathrm{y}(t+\tau))$ at different time delay τ:

$$\mathcal{I}_{\mathrm{xy}}(\tau) = (\mathrm{I}(\mathrm{x}(t);\ \mathrm{y}(t-T)), \ldots, \mathrm{I}(\mathrm{x}(t);\ \mathrm{y}(t+T))), \tag{3}$$

  where $-T \leq \tau \leq T$.

  In general, given a TDMI function $\mathcal{I}_{ij}(\tau)$, one can estimate the time delay $\hat{\tau}_{ij}$ between the ith and jth regions as

$$\hat{\tau}_{ij} = \underset{\tau}{\operatorname{argmax}}\ \mathcal{I}_{ij}(\tau). \tag{4}$$

  By repeating the same process for local activities observed in each pair of decomposed regions, one can construct a time delay matrix $\mathbf{D}$ as follows:

$$\mathbf{D} = [\hat{\tau}_{ij}]_{n \times n}. \tag{5}$$

  The corresponding TDMI matrix is obtained as

$$\hat{\mathrm{I}}_{ij} = \mathcal{I}_{ij}(\hat{\tau}_{ij}), \tag{6}$$

$$\mathbf{I} = [\hat{\mathrm{I}}_{ij}]_{n \times n}. \tag{7}$$

- *Step 2—Generating an optimal dependence tree* (Figs. 1e and 1f)—In this step, the proposed approach finds an optimal dependence tree (Chow-Liu tree [29]) $\mathcal{T}$ that best approximates the graph joint probability $p(\mathbf{X})$ by a product of second-order conditional and marginal distributions. The optimal dependence tree $\mathcal{T}$ can be obtained based on the TDMI matrix $\mathbf{I}$ found in the TDMI analysis. In particular, weights are assigned following $\mathbf{I}$ to each possible edges of a weighted graph with node set $\mathbf{X}$ that encodes no assertion of conditional independence. Prim's algorithm [30] is then applied to find a subset of the edges that forms a tree structure including every node, in which the total weight is maximized.

- *Step 3—Edge orientation* (Figs. 1g and 1h)—The undirected tree $\mathcal{T}$ is transformed to a directed prior graph structure G^{p} by assigning orientations to the edges. Typically, one can assign edge orientations by either selecting a random node as a root node, or by performing a conditional independence test [17] and scoring function optimization over the graph [31]. These methods are either inaccurate or require exhaustive search on all possible edge orientations therefore computationally costly.

  To overcome these problems, we propose to orient the edges by tracing the time delays for each pair of nodes in the tree structure using $\mathbf{D}$ learned by the TDMI analysis (5). In particular, if the activity patterns observed in X_i are lagging the patterns observed in X_j with a time delay τ, it is reasonable to assume that the distribution of X_i is conditionally dependent on X_j. The edge is therefore pointed from X_j to X_i. The direction of an edge with zero time delay is randomly assigned. Those zero delayed edges are found among neighboring regions in the same camera view. It is observed in our experiment that changes of direction in those edges have little effect on the final performance. With G^{p} defined by the edges, one can derive the ordering of variables $\prec$ by performing topological sorting [32]. In particular, the ordering $\prec$ specifies that a variable X_j can only be the parent of X_i if, and only if, X_j precedes X_i in $\prec$, i.e., $X_j \in \mathbf{Pa}(X_i)$ iff $X_j \prec X_i$.

3.3.2 Time Delayed Scored-Searching-Based Learning

In the second stage of the proposed structure learning approach (Fig. 1i), a popular heuristic search method known as the K2 algorithm [14] is reformulated to generate an optimized time delayed dependency structure based on $\prec$ derived from the first-stage learning. Note that without the first-stage learning, one may set $\prec$ randomly. However, a randomly set $\prec$ does not guarantee to give the most probable model structure. Alternatively, one can apply the K2 algorithm exhaustively on all possible orderings to find a structure that maximizes the score. This solution is clearly infeasible even for a moderate number of nodes since the space of ordering is $n!$ for a n-node graph.

Let us now describe the details of the second-stage learning (Algorithm 1). The K2 algorithm iterates over each node X_i that has an empty parent set $\mathbf{Pa}(X_i)$ initially. Candidate parents are then selected in accordance with the node sequence specified by $\prec$ and they are added incrementally to $\mathbf{Pa}(X_n)$, whose addition increases the score of the structure G given data set $\mathcal{X}$. We consider a widely used scoring function that is both score equivalent and decomposable [15], namely, Bayesian Information Criterion score [33]. Specifically, the BIC score is defined as

$$\begin{aligned} S_{\mathrm{BIC}}(G|\mathcal{X}) &= \sum_{i=1}^{n} S_{\mathrm{BIC}}(X_i|\mathbf{Pa}(X_i)) \\ &= \sum_{i=1}^{n}\sum_{t=1}^{m} \log p(x_{i,t}|\mathbf{pa}(X_i), \boldsymbol{\theta}_{x_{i,t}|\mathbf{pa}(X_i)}) \\ &\quad - \log m \sum_{i=1}^{n} \frac{b_i}{2}, \end{aligned} \tag{8}$$

where $b_i = q_i(r_i - 1)$ is the number of parameters needed to describe $p(X_i|\mathbf{Pa}(X_i))$.

Algorithm 1. The reformulated K2 algorithm with a time delay factor being introduced

```
   Input: A graph with a node set $\mathbf{X} = \{X_i | i = 1, 2, \dots, n\}$.
          An ordering of nodes $\prec$. An upper bound $\varphi$ on the
          number the parents a node may have. Time delay
          matrix $\mathbf{D}$.
   Output: Final structure $G$ defined by
          $\{(X_i, \mathbf{Pa}(X_i)) \mid i = 1, 2, \dots, n\}$.
 1 for $i = 1$ to $n$ do
 2     $\mathbf{Pa}(X_i) = \emptyset$;
 3     $score_{\mathrm{old}} = S_{\mathrm{BIC}}(X_i|\mathbf{Pa}(X_i))$;
 4     $OKToProceed = $ true;
 5     while $OKToProceed$ and $|\mathbf{Pa}(X_i)| < \varphi$ do
 6         Let $X_j \prec X_i$, $X_j \notin \mathbf{Pa}(X_i)$, with activity patterns
           $\mathrm{x}_j(t+\tau)$, $\tau = \mathbf{D}(X_i, X_j) \leq 0$, which maximises
           $S_{\mathrm{BIC}}(X_i|\mathbf{Pa}(X_i) \cup \{X_j\})$;
 7         $score_{\mathrm{new}} = S_{\mathrm{BIC}}(X_i|\mathbf{Pa}(X_i) \cup \{X_j\})$;
 8         if $score_{\mathrm{new}} > score_{\mathrm{old}}$ then
 9             $score_{\mathrm{old}} = score_{\mathrm{new}}$;
10             $\mathbf{Pa}(X_i) = \mathbf{Pa}(X_i) \cup \{X_j\}$;
11         else
12             $OKToProceed = $ false;
13         end
14     end
15 end
```

Our formulation differs from the original K2 algorithm in that any addition of a candidate parent is required not only to increase the graph structure score, but it must also satisfy the constraint imposed by the time delays discovered in the first-stage learning. In addition, the score computation (8) is carried out by shifting parent's activity patterns with a relative delay to child node's activity patterns based on $\mathbf{D}$ (see Algorithm 1 (L6)).

3.3.3 Computational Cost Analysis

In this section, the computational cost needed for the proposed two-stage structure learning approach is analyzed. For the first-stage learning (see Section 3.3.1), the total of possible region pairs to be considered for obtaining pairwise TDMI function (3) is in the order of $\mathrm{O}(n^2)$, where n is the number of regions. In each TDMI function computation, if one bounds the maximum time delay to be $\tau_{\max}$, the number of TDMI calculations (2) is $\tau_{\max} - 1$. Hence, the overall complexity of TDMI analysis (Step-1) is $\mathrm{O}(n^2\tau_{\max})$. The runtime complexity of the optimal dependence tree approximation (Step-2) (Section 3.3.1) is $\mathrm{O}(e \log n)$, and the topological sorting (Step-3) takes $\mathrm{O}(n + e)$ time [32], where e is the number of edges.

For the second-stage structure learning (see Algorithm 1), the **for** statement loops $\mathrm{O}(n)$ times. The **while** statement loops at most $\mathrm{O}(\varphi)$ times once it is entered, where φ denotes the maximum number of parents a node may have. Inside the **while** loop, line 6 in Algorithm 1 is executed for at most $n-1$ times since there are at most $n-1$ candidate parents consistent with $\prec$ for X_i. Hence, line 6 in Algorithm 1 takes $\mathrm{O}(sn)$ time if one assumes each score evaluation takes $\mathrm{O}(s)$ time. Other statements in the **while** loop take $\mathrm{O}(1)$ time. Therefore, the overall complexity of the second-stage structure learning is $\mathrm{O}(sn)\mathrm{O}(\varphi)\mathrm{O}(n) = \mathrm{O}(sn^2\varphi)$. In the worst-case scenario where one does not apply an upper bound to the number of parents a node may have, the time complexity becomes $\mathrm{O}(sn^3)$ since $\varphi = n$.

3.3.4 Discussion

Note that both stages of the structure learning method are important to discover and learn the time delayed dependencies among regional activities. Specifically, without the first-stage structure learning, vital time delay information would not be available for constraining the search space. On the other hand, as one shall see later in our experiments (Section 4), poorer results may be obtained if one uses the tree structure alone without the second-stage learning. This is because the tree structure can only approximate an optimum set of $n-1$ first-order dependence relationship among the n variables but not the target distribution, which may include more complex dependencies. Furthermore, studies have shown that the constraint-based learning can be sensitive to failures in independence tests [21]. Therefore, a second-stage scored-searching-based learning is needed to discover additional dependencies and correct potential error in the first-stage learning.

A heuristic search algorithm is chosen for the proposed second-stage structure learning instead of an exact learning algorithm. In general, exact structure learning is intractable for large graph since there are $2^{\mathrm{O}(n^2 \log n)}$ DAGs for an n-node graph [34]. A search using a typical exact algorithm would take exponential time on the number of variables n, e.g., $\mathrm{O}(n2^n)$ for a dynamic programming-based technique [35]. Such high complexity prohibits its use from learning any typical camera network, which may consist of hundreds of local activity regions depending on scene complexity.

Among various heuristic search algorithms, the K2 algorithm [14] is found to be well suited for learning the dependency structure of a large camera network due to its superior computational efficiency. Specifically, thanks to the ordering constraint, the search space of the K2 algorithm is much smaller than that of a conventional GHC search [36]. In addition, the constraint also helps in avoiding costly acyclicity checks since the topological order already ensures the acyclicity of structure. Besides, the K2 algorithm is also more efficient than alternative methods such as Markov Chain Monte Carlo (MCMC)-based structure learning [37], which requires a sufficiently long burn-in time to obtain a converged approximation for a large graph [38].

3.4 Parameter Learning

Parameter learning (Fig. 1l) is performed after we find an optimized structure of the TD-PGM. To learn the parameters of the TD-PGM in a Bayesian learning setting, we use Dirichlet distribution as a conjugate prior for the parameters of the multinomial distribution. The prior of $\boldsymbol{\theta}_{x_i|\mathbf{pa}(X_i)}$ is distributed according to $Dir(\alpha_1, \dots, \alpha_{r_i})$, a posteriori of $\boldsymbol{\theta}_{x_i|\mathbf{pa}(X_i)}$ is updated as $Dir(\alpha_1 + N_{ij1}, \dots, \alpha_{r_i} + N_{ijr_i})$. Here, we apply the BDeu prior (likelihood equivalent uniform Bayesian Dirichlet), $\alpha = \frac{\eta}{r_i q_i}$ over model parameters, where η is known as equivalent sample size [15]. With this Bayesian learning setting, one can update the posterior distribution sequentially and efficiently using a closed-form formula when new instances are observed. To account for a cross-region time delay factor, regional activity patterns are temporally shifted according to the time delay matrix $\mathbf{D}$ during the parameter learning stage.

3.5 Global Unusual Event Detection

A conventional way for detecting unusual events is to examine the log likelihood (LL), $\log p(\mathbf{x}_t|\Theta)$, of the observations given a model, e.g., [11]. Specifically, an unseen global activity pattern is detected as being unusual if

$$\log p(\mathbf{x}_t|\Theta) = \sum_{i=1}^{n} \log p(x_{i,t}|\mathbf{pa}(X_i), \boldsymbol{\theta}_{x_i|\mathbf{pa}(X_i)}) < \mathrm{Th}, \qquad (9)$$

where Th is a predefined threshold and $\mathbf{x}_t = \{x_{i,t}|i = 1, 2, \dots, n\}$ are observations at time slice t for all n regions. However, given a crowded public scene captured using videos with low-image resolution both spatially and temporally, observations $\mathbf{x}_t$ inevitably contain noise and the LL-based method is likely to fail in discriminating the "true" unusual events from noisy observations because both can contribute to a low value in $\log p(\mathbf{x}_t|\Theta)$, and thus cannot be distinguished by examining $\log p(\mathbf{x}_t|\Theta)$ alone.

We address this problem by introducing a Cumulative Abnormality Score (CAS) that alleviates the effect of noise by accumulating the temporal history of the likelihood of unusual event occurrences in each region over time. This is based on the assumption that noise would not persist over a sustained period of time and thus can be filtered out when visual evidence is accumulated over time. Specifically, an abnormality score (set to zero at $t = 0$) is computed for each node in the TD-PGM on-the-fly to monitor the likelihood of abnormality for each region. The log likelihood of a given observation $x_{i,t}$ for the ith region at time t is computed as

$$\begin{aligned} &\log p\big(x_{i,t}|\mathbf{pa}(X_i), \boldsymbol{\theta}_{x_{i,t}|\mathbf{pa}(X_i)}\big) \\ &\quad = \log \frac{N_{x_{i,t}|\mathbf{pa}(X_i)} + \frac{\eta}{r_i q_i}}{\sum_{k=1}^{r_i} \Big(N_{x_{i,t}=k|\mathbf{pa}(X_i)} + \frac{\eta}{r_i q_i}\Big)}. \end{aligned} \qquad (10)$$

If the log likelihood is lower than a threshold Th_i, the abnormality score for $x_{i,t}$, denoted as $c_{i,t}$, is increased as: $c_{i,t} = c_{i,t-1} + |\log p(x_{i,t}|\mathbf{pa}(X_i), \boldsymbol{\theta}_{x_i|\mathbf{pa}(X_i)}) - \mathrm{Th}_i|$. Otherwise, it is decreased from the previous abnormality score: $c_{i,t} = c_{i,t-1} - \delta(|\log p(x_{i,t}|\mathbf{pa}(X_i), \boldsymbol{\theta}_{x_i|\mathbf{pa}(X_i)}) - \mathrm{Th}_i|)$, where δ is a decay factor controlling the rate of the decrease. $c_{i,t}$ is set to 0 whenever it becomes a negative number after a decrease. Therefore, $c_{i,t} \geq 0$, $\forall\{i, t\}$, with a larger value indicating higher likelihood of being unusual. Note that during the computation of log likelihood (10), the activity patterns of a parent node are referred to based on the relative delay between the parent node and the child node.

A global unusual event is detected at each time frame when the total of CAS across all the regions is larger than a threshold Th, that is,

$$C_t = \sum_{i=1}^{n} c_{i,t} > \mathrm{Th}. \tag{11}$$

Overall, there are two thresholds to be set for global unusual event detection. Threshold Th_i is set automatically to the same value for all the nodes as $\overline{LL} - \sigma^2_{LL}$, where the $\overline{LL}$ and σ^2_{LL} are the mean and variance of the log likelihoods computed over all the nodes for every frames, which are obtained from a validation data set. The other threshold Th is set according to the detection rate/false alarm rate requirement for specific application scenarios.

Once a global unusual event is detected, the contributing local activities of individual regions can be localized by examining $c_{i,t}$. Particularly, $c_{i,t}$ for all regions are ranked in a descending order. Local activities that contribute to the unusual event are then identified as those observed from the first few regions in the rank that are accounted for for a given fraction $P = [0, 1]$ of C_t.

3.6 Incremental Two-Stage Structure Learning

As discussed in Section 1, incremental learning is needed to cope with visual context changes over time. In contrast to a batch-mode learning method that performs single-round learning using a full training set, an incremental learning method outputs a model at each time point based on a stream of observations. Formally, given a new observation $\mathbf{x}_t$ at each time step t, an incremental graphical model learning method produces a model B_t with a refined structure G_t and the associated parameters Θ_t. In practice, the incremental learning process may only be invoked after collecting some number of h instances.

For incremental structure learning, one can employ a **Naive** method in which all the observations seen so far, $\mathbf{x}_1, \ldots, \mathbf{x}_t$, are used to estimate G_t. Obviously, the method should yield an optimal structure since all the observed information is used for the estimation. The method, however, is memory prohibitive because it needs to either store all the previously seen instances or keep a count of the number of times each distinct instantiation to all variables $\mathbf{X}$ is observed.

Alternatively, one can approximate a maximum a posteriori probability (MAP) model [18], [39], i.e., a model that is considered most probable given the data seen so far. All the past observations can be summarized using the model, which is then exploited as a prior in the next learning iteration for posterior approximation. The **MAP** approach is memory efficient because it only needs to store new instances that one has observed since the last MAP update. This method, however, may lead to poor incremental learning since subsequent structures can be easily biased to the initial model [18].

Unlike **Naive** and **MAP**, our incremental structure learning method takes constant time regardless the number of instances observed so far, and it is memory tractable without sacrificing the accuracy of the structure learned. Importantly, our method employs a constant time window-based incremental learning to ensure that the time evolution of behavior is captured and constantly updated in the graphical model.

The steps involved in the proposed incremental structure learning method are summarized in Algorithm 2. In the proposed approach, an obsolete structure is replaced by searching from a set of most probable candidate structures at the current time, which are stored in a frontier $\mathcal{F}$ [18]. The associated sufficient statistics ξ of $\mathcal{F}$ are kept to allow tractable update of model parameters via Bayesian learning. Note that the structure learning process is invoked after receiving h instances, $\mathbf{x}_{t-h+1\,:\,t}$, to ensure sufficient information for learning the TDMI functions. In addition, there must be at least half of the h instances scoring below a predefined filtering threshold $\mathrm{Th}_{\mathrm{CAS}}$ during unusual event detection (Algorithm 2 (L3-4)). The filtering step is introduced to prevent excessive number of outliers from being incorporated inadvertently into the model updating process. Similarly to Th_i (Section 3.5), the threshold $\mathrm{Th}_{\mathrm{CAS}}$ is obtained from a validation set. Specifically, after we obtain Th_i, we compute C_t for every frames and set $\mathrm{Th}_{\mathrm{CAS}}$ as

$$\frac{\sum_{t=1}^{l} C_t}{2l},$$

where l is the total frames of the validation data set.

Algorithm 2. Incremental two-stage structure learning

```
Input: Data stream $(\mathbf{x}_1, \ldots, \mathbf{x}_t, \ldots)$. An upper bound, $\varphi$,
       on the number the parents a node may have.
       Number of past instances to keep, $h$. An initial
       model structure, $G_0$. A set of sufficient statistics,
       $\boldsymbol{\xi}_0 = \xi(G_0)$. Update coefficient $\beta$.
Output: $G_t$ and $\boldsymbol{\xi}_t$.
1  for t from $1, 2, \ldots$ do
2      $G_t = G_{t-1}$, $\boldsymbol{\xi}_t = \boldsymbol{\xi}_{t-1}$;
3      Receive $\mathbf{x}_t$. Compute $C_t$ [(11)] ;
4      if $t \bmod h = 0$ and
       $|\{C_i | t-h+1 \leq i \leq t,\, C_i < \mathrm{Th}_{\mathrm{CAS}}\}| \geq \frac{h}{2}$ then
           // Stage One
5          Compute $\mathcal{I}(\tau)$ using $\mathbf{x}_{t-h+1:t}$ [(2) and (3)] ;
6          if $t = 1$ then
7              Set $\mathcal{I}^{\mathrm{acc}}(\tau) = \mathcal{I}(\tau)$ ;
8          else
9              Update $\mathcal{I}^{\mathrm{acc}}(\tau)$ with updating rate $\beta$ [(12)] ;
10         end
11         $\overline{\mathcal{I}}^{\mathrm{acc}}(\tau) = \mathcal{I}^{\mathrm{acc}}(\tau)$ ;
12         Compute $\mathbf{D}$ and $\mathbf{I}$ using $\mathcal{I}^{\mathrm{acc}}(\tau)$ [(4) to (7)] ;
13         Find the ordering of variables $\prec$ [Sec. 3.3.1] ;
           // Stage Two
14         Create $\mathcal{F}$ based on $\prec$ and $G_{t-1}$ ;
15         Obtain $\boldsymbol{\xi}_t$ by updating $\boldsymbol{\xi}_{t-1}$ using $\mathbf{x}_{t-h+1:t}$ ;
16         Search for highest scored $G_t$ from $\mathcal{F}$ [Alg. 1];
17     end
18 end
```

Let us now detail the steps involved in the proposed incremental structure learning:

- *Step 1—Finding a topological order* $\prec$ —Similarly to the batch-mode learning described in Section 3.3, there are two stages in our incremental structure learning procedure. The learning process commences with the estimation of ordering of variables $\prec$ in the first-stage learning (Algorithm 2 (L5-13)).

  In particular, up-to-date cumulative TDMI functions $\mathcal{I}^{\mathrm{acc}}(\tau)$ for each pair of regional activity patterns are first estimated by accumulating past

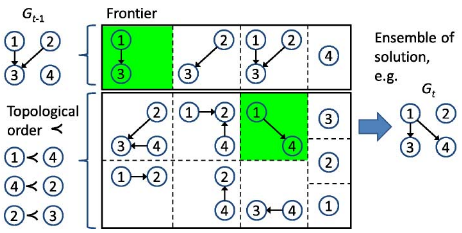


Fig. 2. The frontier, $\mathcal{F}$, is constructed based on G_{t-1} and $\prec$, forming a set of *families* composed of X_i and its parent sets $\mathbf{Pa}(X_i)$. The new structure G_t is an example of an implicit ensemble of solutions that can be composed of every possible combination of the families.

TDMI functions. Specifically, $\mathcal{I}^{\text{acc}}_{ij}(\tau)$ between the ith region and jth region is estimated as follows:

$$\mathcal{I}^{\text{acc}}_{ij}(\tau) = \beta\overline{\mathcal{I}}^{\text{acc}}_{ij}(\tau) + (1-\beta)\mathcal{I}_{ij}(\tau), \qquad (12)$$

where β denotes an update coefficient that controls the updating rate of the function, $\overline{\mathcal{I}}^{\text{acc}}_{ij}(\tau)$ represents the cumulative TDMI function found in previous learning iteration, and $\mathcal{I}_{ij}(\tau)$ denotes a TDMI function computed using $\mathbf{x}_{t-h+1\,:\,t}$. Given $\mathcal{I}^{\text{acc}}_{ij}(\tau)$, one can obtain the updated $\mathbf{I}$, $\mathbf{D}$, and $\prec$ using the procedures described in Section 3.3.1.

- *Step 2—Building a frontier* $\mathcal{F}$—After obtaining G^{p} and $\prec$, we proceed to the second-stage learning. We first construct a frontier $\mathcal{F}$ based on $\prec$ and a structure estimated in previous iteration, G_{t-1} (Algorithm 2 (L14), see Fig. 2 for an illustration). Formally, $\mathcal{F}$ is defined by a set of *families* composed of X_i and its parent set $\mathbf{Pa}(X_i)$:

$$\mathcal{F} = \{(X_i, \mathbf{Pa}_j(X_i)) \mid 1 \le i \le n,\ 1 \le j \le \Omega\}, \qquad (13)$$

where Ω denotes the total number of different parent sets $\mathbf{Pa}_j(X_i)$ associated with X_i.

We construct $\mathcal{F}$ by including existing families in G_{t-1} as well as using different combinations of candidate parents of X_i consistent with $\prec$. With this strategy, one could build a new structure G_t that could be simpler or more complex than G_{t-1} through combining different families in $\mathcal{F}$ (Fig. 2).

To prevent proliferation of parent set combinations and to constrain the search space to a set of most promising structures for incremental learning, we *prune less probable candidate structures* from joining the final scoring process. In particular, different combinations of parent set for X_i are formed by selecting only a set of most probable parents, $\mathbf{mpp}_i$ consistent with $\prec$, with $|\mathbf{mpp}_i| \le \varphi < n$. Here, φ denotes the maximum number of parents a node may have and $\mathbf{mpp}_i$ contains parents that return the highest TDMI among other candidate parents. The maximum number of parent combinations a node may have is given as $\Omega = 1 + \sum_{k=1}^{\varphi}\binom{\varphi}{k}$.

- *Step 3—Updating sufficient statistics* $\boldsymbol{\xi}$—Since $\mathcal{F}$ at time t may be different from that at $t-1$, one needs to update the associated sufficient statistics of each family in $\mathcal{F}$ to quantify the most recent multinomial CPDs (Algorithm 2 (L15)). A set of such sufficient statistics at time t is denoted as $\boldsymbol{\xi}_t$. Given $\mathcal{F}$, $\boldsymbol{\xi}_t$ is obtained from previous set of sufficient statistics $\boldsymbol{\xi}_{t-1}$ and the h recent instances $\mathbf{x}_{t-h+1:t}$ as follows:

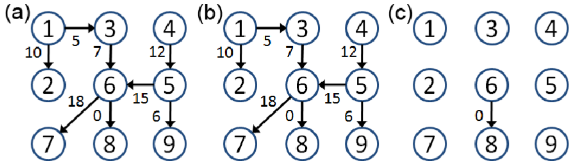


Fig. 3. (a) Ground truth structure, (b) structure learned using the proposed two-stage structure learning, and (c) structure learned using the conventional K2 algorithm without time delay estimation.

$$\boldsymbol{\xi}_t = \boldsymbol{\xi}_{t-1} \bigcup \{\mathbf{N}_{X_i|\mathbf{Pa}(X_i)} \mid (X_i, \mathbf{Pa}(X_i)) \in \mathcal{F}\}, \qquad (14)$$

where $\mathbf{N}_{X_i|\mathbf{Pa}(X_i)} = \{N_{x_i|\mathbf{pa}(X_i)}\}$, with $N_{x_i|\mathbf{pa}(X_i)}$ being the number of cases $(x_i, \mathbf{pa}(X_i))$ observed in the h recent instances, $\mathbf{x}_{t-h+1\,:\,t}$. The updated sufficient statistics $\boldsymbol{\xi}_t$ will then be used in the next step for structure scoring. Note that after the incremental two-stage structure learning, $\boldsymbol{\xi}_t$ will also be used for parameter update via Bayesian learning as described in Section 3.4.

- *Step 4—Scoring a structure*—In this step, we wish to search for an optimal structure G_t within $\mathcal{F}$ to replace G_{t-1} (Algorithm 2 (L16)). This is achieved by comparing the scores returned from a set of candidate structures that can be evaluated using the records in $\boldsymbol{\xi}_t$, that is:

$$G_t = \operatorname*{argmax}_{\{G' \mid \xi(G')\subset\boldsymbol{\xi}_t\}} S^*_{\text{BIC}}(G'|\boldsymbol{\xi}_t), \qquad (15)$$

where $S^*_{\text{BIC}}(\cdot)$ denotes a modified version of the original score $S_{\text{BIC}}(\cdot)$ defined in (8).

As pointed out by Friedman and Goldszmidt [18], the score needs to be modified because one may start collecting new sufficient statistics or may remove redundant ones at different times, due to addition/removal of families from $\mathcal{F}$ during the incremental structure learning. The number of instances $\mathbf{N}_{X_i|\mathbf{Pa}(X_i)}$ recorded in a family's sufficient statistics would affect the final score value, e.g., a lower score may be assigned to a family that observes more instances. To avoid unfair comparison of different candidate structures, it is thus necessary to average the score yielded by each family with the total instances recorded in its sufficient statistics. In particular, this work follows the method proposed by Friedman and Goldszmidt [18] to modify S_{BIC}:

$$S^*_{\text{BIC}}(X_i|\mathbf{Pa}(X_i)) = \frac{S_{\text{BIC}}(X_i|\mathbf{Pa}(X_i))}{\sum_{(x_i|\mathbf{pa}(X_i))} N_{x_i|\mathbf{pa}(X_i)}}. \qquad (16)$$

Since the proposed method includes the previous graph structure G_{t-1} in $\mathcal{F}$ and its sufficient statistics in every learning iteration, the incremental learning procedure will improve monotonically as it must return a structure G_t that scores at least as well as G_{t-1}, i.e., $S^*_{\text{BIC}}(G_t|\boldsymbol{\xi}) \ge S^*_{\text{BIC}}(G_{t-1}|\boldsymbol{\xi})$.

4 Experimental Results

4.1 Synthetic Data

We first demonstrate some of the advantages and important properties of the proposed method through analyzing a

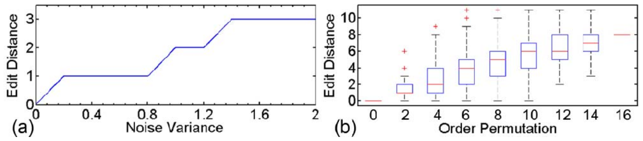


Fig. 4. Edit distance against (a) white noise and (b) order permutation.

synthetic data set—a nine-node network with known structure, internode delays, and CPDs (Fig. 3a). For each of the nodes, we generated time series with a 10 state-levels and a length of 10,000 from the synthetic network.[2]

In the first experiment, we compared the structure learned using the proposed two-stage structure learning with that learned using the conventional K2 algorithm without time delay estimation. As can be seen from Fig. 3b, the proposed method recovered all the edges including the correct internode delays. In comparison, the conventional K2 algorithm can only discover the zero delayed edge, but failed to infer other time delayed edges.

In the second experiment, the learned structure was evaluated when an increasing amount of random white noise was added to the training data. The accuracy of structure learning was measured by "edit distance," defined by the length of the minimal sequence of operators needed to transform the ground truth structure into the resulting one (operators are edge-insertion, edge-deletion, and edge-reversal). The result depicted in Fig. 4a suggests that the method is robust to a low level of noise but could produce false edges when the data are saturated by noise.

In the third experiment, we examined the final structure produced by the K2 algorithm by feeding it with an exhaustive set of node order with different permutations. The experimental result (Fig. 4b) suggests that a less accurate initial estimate could lead to a poor final structure. However, the K2 algorithm is capable of mitigating the negative effect if there is only a minor permutation on the node order. This can be seen from the low sample minimum and lower quartile of the box plots even when the order permutation was increased to 10.

4.2 Batch Mode Structure Learning

In Sections 4.2 and 4.3, we employed a challenging multicamera data set[3] that contains fixed views from nine disjoint and uncalibrated cameras installed at a busy underground station (Fig. 5). Three cameras were placed in the ticket hall and two cameras were positioned to monitor the escalator areas. Both train platforms were covered by two cameras each. The video from each camera lasts around 19 hours from 5:42 am to 00:18 am the next day, giving a total of 167 hours of video footage at a frame rate of 0.7 fps. Each frame has a size of 320×230.

Passengers typically enter from the main entrance, walk through the ticket hall or queue up for tickets (Cam 1), enter the concourse through the ticket barriers (Cam 2, 3), take the escalators (Cam 4, 5), and enter one of the platforms. The opposite route is taken if they are leaving the station. The data set is challenging due to

1. complex crowd dynamics,
2. the complexity and diversity of the scene,
3. low video temporal and spatial resolution,
4. the enormous number of objects appearing in the scene, and
5. the existence of multiple entry and exit points which are not visible in the camera views.

The data set was divided into 10 subsets, each of which contains 5,000 frames per camera (≈ 2 hours in length, the last subset contains 1,500 frames). Two subsets were used as validation data. For the remaining eight subsets, 500 frames/camera from each subset were used for training and the rest for testing, i.e., 10 percent for training.

4.2.1 *Global Activity Dependency Modeling*

Using the training data, the nine camera views were automatically decomposed into 96 semantically meaningful regions (Fig. 5). Given the scene decomposition, the global activities, composed of 96 regional activities, were modeled using a TD-PGM. The model structure, which encodes the time delayed dependencies among regional activities, was initialized using pairwise TDMI and then optimized using the proposed two-stage structure learning method. The structure yielded is depicted in Fig. 6.

As expected, most of the discovered dependencies were between regions from the same camera views that have short time delays (Fig. 6). However, a number of interesting dependencies between intercamera regions were also discovered accurately. For instance, three escalator entry and exit zones in Cam 4 (Regions 40, 43, and 46) were found to be connected with individual escalator tracks in Cam 5 (Regions 55, 59, and 51) despite the fact that some of the connecting regions were not visible in the camera views. Importantly, correct directions of edge dependency were also discovered, e.g., an edge pointing from the upward escalator track (Region 51) toward the corresponding exit zone (Region 46). The interregional time delays estimated were also very close to the time gap manually observed, e.g., 8, 5, and 10 frames for region pairs 40-55, 43-59, 46-51, respectively.

We compared our method ($TDMI{+}K2$) with three alternative dependency learning methods:

1. $MI{+}K2$—The proposed two-stage structure learning method but initialized using MI rather than TDMI, to demonstrate the importance of encoding time delay.
2. $TDMI$—First-stage structure learning only, to highlight the importance of having two stages in structure learning.
3. $xCCA{+}K2$—The proposed structure learning method but initialized using pairwise Cross Canonical Correlation Analysis (xCCA) proposed in [23] rather than TDMI, to show the advantage of modeling nonlinearity among activity dependencies using TDMI for global unusual event detection.

Note that the same global activity representation described in Section 3.1 was applied on both the TDMI and xCCA-based approaches.

The dependency structures discovered by the proposed method and the three alternative approaches are depicted in Fig. 7, with some critical missed/incorrect dependency links highlighted with red squares. As one shall see in the

[2] To generate a sample, we drew the parent node values randomly and generated the child node values following the network parameters.

[3] Processed data are available at http://www.eecs.qmul.ac.uk/~ccloy/files/datasets/liv_processed.zip.

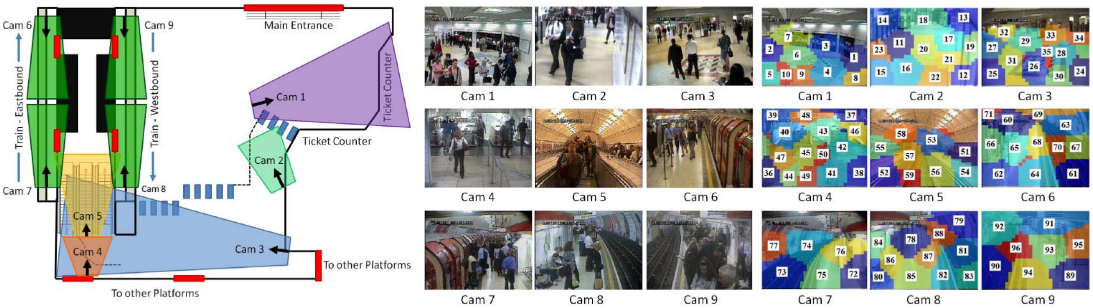


Fig. 5. The underground station layout, the camera views, and the scene decomposition results for our data set. Entry and exit points are shown in red bars.

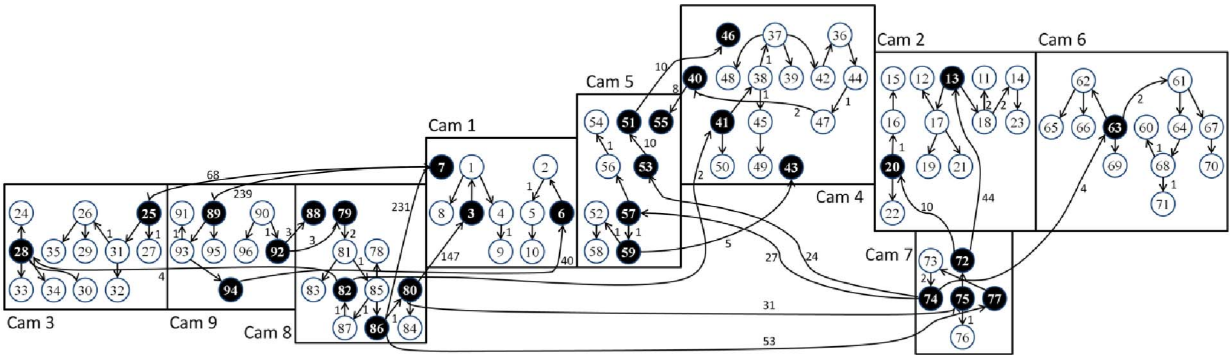
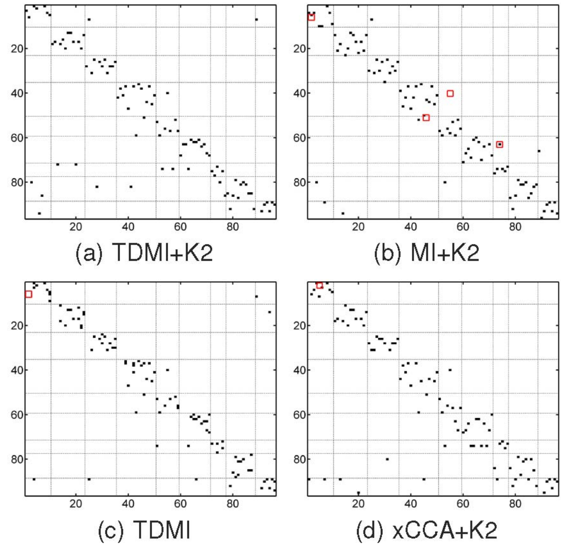


Fig. 6. An activity global dependency graph learned using the proposed two-stage structure learning method with BIC scoring function. Edges are labeled with the associated time delays discovered using the Time Delayed Mutual Information analysis. Regions and nodes with discovered intercamera dependencies are highlighted.

unusual event detection experiment (see Section 4.2.2), these links play an important role in unusual event detection. From Fig. 7b, it is observed that without taking time delay into account, $\mathrm{MI}+\mathrm{K2}$ yielded a number of missed dependency links such as $40 \rightarrow 55$ and $51 \rightarrow 46$, as well as incorrect one such as $63 \rightarrow 74$, which were against the causal flow of activity patterns. Fig. 7c shows that without global dependency optimization, structure yielded by TDMI alone was inferior to that obtained using TDMI+K2. In particular, some important dependency links such as $6 \rightarrow 2$ were still not discovered. It is observed that some links such as $2 \rightarrow 5$ was missing when we initialized the proposed structure learning method using xCCA (Fig. 7d). This is due to the use of pairwise linear correlations without taking into account nonlinearity among activity dependencies across regions.



(a) TDMI+K2

(b) MI+K2

(c) TDMI

(d) xCCA+K2

Fig. 7. Activity global dependency structures learned using different methods. The *y-axis* represents the parent nodes, while the *x-axis* represents the child nodes. A black mark at (y,x) means $y \rightarrow x$. Some missed or false edges are highlighted using squares in red.

4.2.2 *Unusual Event Detection*

For quantitative evaluation of our unusual event detection method, ground truth was obtained by an exhaustive frame-wise examination on the entire test set. Consequently, nine unusual cases were found, each of them lasting between 34 to 462 frames, with an average of 176 frames (254 secs). In total, there were 1,585 atypical frames accounting for 4.88 percent of the total frames in the test set. As shown in Table 1, these unusual cases fall into three categories, all of which involve multiple regional activities.

A TD-PGM learned using our TDMI+K2 method was employed for unusual event detection using the proposed CAS. The decay factor δ of CAS was found to produce consistent results when it is set beyond value 5. Consequently, it was set to 10 throughout the experiments. For all experiments, a weak uniform prior $\eta = 10$ was used. The performance of the proposed approach ($TDMI{+}K2{+}CAS$) was assessed using a receiver operating characteristic (ROC) curve by varying the other free parameter threshold Th. An unusual event is considered detected if and only if its CAS is $>\mathrm{Th}$ and at least half of the detected regions are consistent with the manually labeled regions in the ground truth.

CAS versus LL—We first examine how effective the proposed CAS is for unusual event detection. Specifically, our approach was compared with a method that used the

TABLE 1
Ground Truth

Cases	Unusual Event Description	Cam	Total frames (% from total)
1-6	The queue in front of the ticket counters was built to a sufficient depth in regions 2 and 6 that it blocked the normal route from Region 2 to 5 taken by passenger who did not have to buy ticket (Fig. 10)	1	1021 (3.14)
7-8	Faulty train observed in Cam 6 and 7 led to overcrowding on the platform. To prevent further congestion on the platform, passengers were disallowed to enter the platform via the escalator (Region 55 in Cam 5). This in turn caused congestion in front of the escalator entry zone in Cam 4 (Fig. 11)	3, 4, 5, 6, 7	446 (1.37)
9	Train moved in reversed direction	6,7	118 (0.36)

same TD-PGM but with the conventional LL score, denoted as TDMI+K2+LL. As can be seen from Fig. 8, using the LL-based abnormality score, the true unusual events were overwhelmed by the noise collected from the large number of regions and thus difficult to detect. Since there was an excessive number of regions falsely identified as atypical, TDMI+K2+LL essentially gave zero true positive rate across all Th; its performance is thus not available to be shown in Fig. 9. In contrast, the proposed CAS effectively alleviated the effect of noise, thus offering much more superior unusual event detection performance.

TDMI+K2+CAS versus other learning methods—We further investigate how unusual event detection performance can be affected when the time delayed dependency structure are not learned accurately. More specifically, TD-PGMs were learned using MI+K2, TDMI alone without second-stage learning, and xCCA+K2, respectively, as described in Section 4.2.1. Cumulative abnormality score was then used for unusual event detection. These three methods are denoted as MI+K2+CAS, TDMI+CAS, and xCCA+K2+CAS, respectively. It is observed from Fig. 9 that without accurate dependencies learned using the proposed TDMI+K2, all three methods yielded poorer performance. In particular, the missing time delayed dependencies shown in Fig. 7 caused missed detection or weak response to unusual events.

Besides the K2 algorithm, we also reformulated another popular scored-searching method known as greedy hill-climbing search [20] for learning time delayed dependencies on the same data set. Slightly poorer performance was obtained, with area under ROC (AUROC) of 0.7558 compared to 0.8458 obtained using TDMI+K2. The poorer detection performance of GHC method was mainly due to its weaker responses on atypical long queue events. In addition, we also followed the method proposed in [11] to construct a CHMM with each chain corresponded to a region. However, the model is computationally intractable on the machine employed in this study (single-Core 2.8 GHz with 2 G RAM) due to the high space complexity during the inference stage.

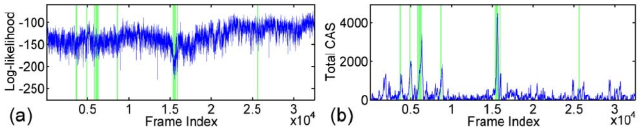


Fig. 8. Unusual event scores computed using (a) log likelihood and (b) cumulative abnormality score. Ground truths of unusual events are represented as bars in green color.

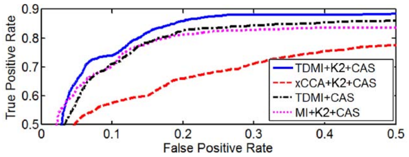


Fig. 9. Receiver operating characteristic curves obtained using the time delayed probabilistic graphical model with different learning methods.

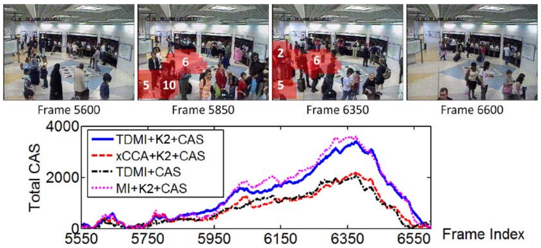


Fig. 10. Example frames from detection output using the proposed approach on analyzing unusual events caused by atypical long queues. The plot depicts the associated cumulative abnormality scores produced by different methods over the period. In ground truth, unusual events occurred at frames (5,741-5,853) and (5,915-6,376).

An example of a detected unusual event using the proposed TDMI+K2+CAS approach is given in Fig. 10. The contributing atypical regions are highlighted in red, following the method described in Section 3.5, with $P = 0.8$. The atypical long queue was robustly detected using the proposed solution. In comparison, other methods such as TDMI+CAS and xCCA+K2+CAS yielded a weaker response. Note that MI+K2+CAS was able to detect this unusual event since the event occurred within a single view, of which the time delays between regional activities can be ignored. One shall see in the next example that MI+K2+CAS failed in detecting a global unusual event that took place across multiple camera views.

Another example of unusual event detection using the proposed approach is shown in Fig. 11. This event was Case 7 listed in Table 1. As can be seen, TDMI+K2+CAS detected the unusual event across Cam 3, 4, 5, 6, and 7 successfully. Specifically, TD-PGM first detected atypical crowd dynamics in Cam 6 and Cam 7, i.e., all train passengers were asked to get off the train. From frame 15,340 to frame 15,680, passengers were disallowed to use the downward escalator and therefore started to accumulate at the escalator entry zone in Cam 4. The congestion led to high CAS in several regions in Cam 4 and Cam 3 (Region 32). A large volume of crowd in Region 55 of Cam 5 was expected due to the high crowd density in Region 40 of Cam 4. However, the fact that Region 55 was empty violated the model's expected time delayed dependency, therefore causing a high CAS in Region 55 (Fig. 11). Despite the event in Region 55 appearing perfectly normal when examined in isolation, it was successfully detected as being unusual since the proposed approach associated Region 55 with Region 40, which has an immediate and direct causal effect on it (Fig. 6). In contrast, MI+K2+CAS failed to discover the time delayed dependencies between Region 40 and Region 55; it therefore missed the unusual event in Region 55. In this example (Fig. 11), xCCA+K2+CAS yielded a

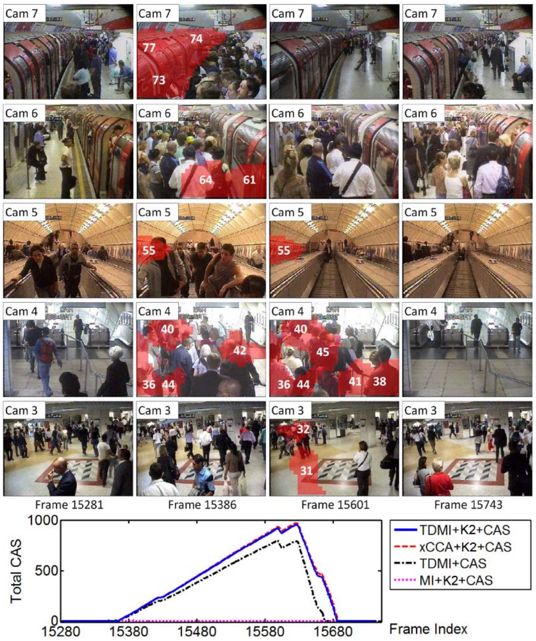


Fig. 11. Global unusual event due to faulty train, which first occurred in Cam 6 and 7, and later propagated to Cam 5, 4, and 3. The plot depicts the cumulative abnormality scores in Region 55 produced by different methods over the period. In ground truth, this unusual event occurred at frames (15,340-15,680).

similar response to that obtained using the proposed approach. However, it is observed from the ROC curve (Fig. 9) that the overall performance of xCCA+K2+CAS was still inferior to that of TDMI+K2+CAS.

4.3 Incremental Structure Learning

4.3.1 Gradual Context Change

This experiment was similar to the global unusual event detection experiment reported in Section 4.2.2. In this experiment, however, a model was no longer trained using data subsets obtained from different time periods, but initialized using only training data extracted in the morning (5:42 am-9:42 am) and updated using subsequent observations using an incremental structure learning method. The goal of this experiment is to compare the proposed incremental structure learning approach (**Incremental**, described in Algorithm 2) with three alternative strategies in dealing with gradual context changes, e.g., crowd flow transitions at different time periods. The three methods were:

1. **ParamAlone**—This method only update parameters alone without adapting the structure of a model.
2. **Naive**—This method stores all past observations for incremental structure learning (described in Section 3.6).
3. **MAP**—This method [18], [39] uses the best model so far as a prior for subsequent structure learning (described in Section 3.6).

We evaluated the aforementioned methods on the underground data set described at the beginning of Section 4. Specifically, two subsets in the morning period were used to initialize a model. A subsequent subset was reserved as validation data to compute the thresholds Th_i and $\mathrm{Th}_{\mathrm{CAS}}$. Other subsets were employed for testing and incremental learning. For all methods, the following settings were used: a slow updating rate with update coefficient, $\beta = 0.9$, and an upper limit of the number of parents a node may have, $\varphi = 3$. All incremental structure learning approaches generated an updated model by invoking the TDMI+K2 structure learning along with individual incremental learning scheme, together with the parameter learning every time $h = 500$ instances were observed. After each learning iteration, the updated model was employed for unusual event detection on subsequent observations. **Naive** and **Incremental** were carried out with the BIC score and the modified BIC score (16), respectively. The BIC score, however, is not suitable for the **MAP** method if one wishes to take into account the prior information represented in a MAP model. Therefore, a modified BDeu score [18] was employed for **MAP**.

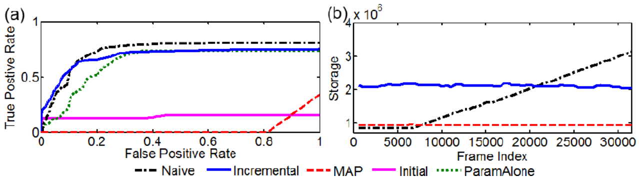


Fig. 12. (a) Receiver operating characteristic curves obtained using different incremental structure learning methods. (b) Memory requirement of different incremental structure learning methods.

Similar to the experiment reported in Section 3.5, the performance of an approach was assessed using ROC curve, which was generated by varying the threshold Th. The ROC curves yielded by a baseline method **Initial** (i.e., an initial model was used without any structure/parameter update), **ParamAlone**, **Naive**, **Incremental**, and **MAP** are shown in Fig. 12a. The memory requirement[4] associated with different incremental structure learning methods is also given in Fig. 12b. Poor detection performance ($\mathrm{AUROC} = 0.1413$) of **Initial** is expected since the initial model only accessed observations in the morning period, which was quiet most of the time. It therefore failed to cope with the busier context in the subsequent subsets. Among three incremental structure learning approaches, we found that **Naive** yielded the best unusual event detection performance, with AUROC of **0.7303**. However, its memory requirement increased linearly along with the number of observations seen, as depicted in Fig. 12b. Despite **MAP** needing the least memory, it was trapped in a wrong structure and subsequently locked to it, yielding the poorest result ($\mathrm{AUROC} = \mathbf{0.0323}$) among all methods. Overall, **Incremental** gave comparable detection performance compared to **Naive**, with an AUROC of **0.6851**. Importantly, memory required by **Incremental** remained constant throughout the test by keeping a handful of sufficient statistics (Fig. 12b). In comparison to **Naive** and **Incremental**, inferior performance was observed on **ParamAlone**, with an AUROC of **0.6278**. This suggests that updating parameters alone may still be inadequate when dealing with gradual visual context changes. Nonetheless, it was still better than maintaining the fixed model's parameters without incremental update.

[4] This study estimates of memory usage of **Naive** and **MAP** based on the number of instances to keep. For **Incremental**, the memory was measured based on the space needed to store the sufficient statistics.

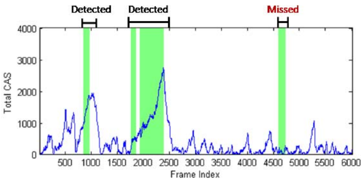


Fig. 13. Abnormality in an online setting can depend on the relative frequency of the events at different time periods. The figure shows a long queue event is detected as unusual at first, but becomes normal later as evidence accumulates. Ground truths of the long queue events are represented as bars in green color.

Note that the long queue event in Cam 1 (Fig. 10) was rare over the whole training set, but a careful inspection shows that it took place rather frequently in certain periods of time of the day (e.g., morning rush hours), so much so that it should be considered as normal during those periods. It is found that the incremental learning method can correctly adapt to the variations and detect it when it is rare and ignore it when it occurs frequently (Fig. 13). The lower AUROC of the **Incremental** strategy compared to the **Naive** strategy should not be interpreted as poorer performance in this context. It was mainly caused by how the ground truth was set. Specifically, it is difficult to accommodate the changes of definition of abnormality/normality in the ground truth as those changes are rather subjective to quantify. Consequently, we assume a fixed definition of abnormality/normality in the ground truth, e.g., a long queue was consistently labeled as abnormal. Therefore, when the incremental learning method ignores the long queue event during rush hours of the day, it leads to miss-detections according to the "assumed constant" ground truth. One therefore expects that the **Incremental** strategy will yield a better performance when the changing definition of abnormality/normality is reflected in the ground truth.

To give further insights, the differences between the initial model and the up-to-date model induced using **Incremental** were investigated. With the proposed incremental structure learning, it was observed that some errors in the initial model were corrected, e.g., initial dependency link $45 \rightarrow 55$ was corrected to $40 \rightarrow 55$, in which Region 40 has a more direct causal impact to Region 55 (Fig. 5). In another example, incorrect time delay estimated between two neighboring Regions 6 and 7 was also corrected from 34 to 2 frames.

The proposed incremental learning approach also learned meaningful changes of interregional dependency strength over time. In an example shown in Fig. 14, since passengers mostly commuted to the city center in the morning/afternoon periods, Region 82 (westbound platform toward city center) thus exhibited a stronger dependency with Region 55 (downward escalator to platforms), as compared to Region 75 (eastbound platform toward residential areas). However, this scenario changed in the late afternoon/evening when people began to travel back home. Hence, the eastbound platform became busier than the westbound platform, started around frame 20,000, i.e., 6-7 pm. The westbound platform remained busy as many commuters took a train from this platform to transit to other stations; thus it still maintained a strong connection with Region 55. It is observed from Fig. 14 that the proposed incremental learning method was able to capture this dependency transition.

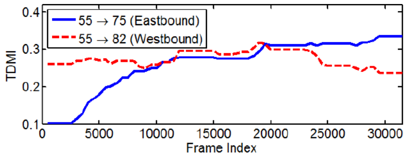


Fig. 14. Interregional dependency changes captured using the proposed incremental two-stage learning.

4.3.2 Abrupt Context Change

In this experiment, we wish to evaluate how well an incrementally trained model can adapt to visual context undergoing abrupt changes. Two scenarios were tested: removal and addition of cameras in a camera network. In the first scenario, all observations from Cam 5 were discarded starting from frame 7,500 to simulate a faulty camera or removal of camera. The second scenario began with eight cameras, and Cam 5 was attached to the network starting from frame 7,500.

The goodness of adaptation was evaluated using a standard measure of density estimation performance, known as *log-loss* [40], which is defined as $l_{\text{loss}} = \frac{1}{m}\sum_{t=1}^{m} \log p(\mathbf{x}_t|\Theta)$, where m is the total number of test cases. In this experiment, l_{loss} was further divided using the total number of decomposed regions in a camera network so that a fair comparison can be performed between two structures with different number of cameras and decomposed regions. The computation of log-loss requires an independent set of test samples. Consequently, we applied a different training/testing data partitioning strategy as employed in Section 4.3.1. In particular, recall that the underground data set was divided into 10 subsets; 2,500 frames from each subset (excluding the three subsets employed for initialization and validation) were used for incremental structure learning and the remaining samples in a subset were reserved for log-loss computation. Similarly to Section 4.3.1, all approaches invoked structure learning and parameter learning every $h = 500$ instances were observed.

The results on both scenarios are depicted in Figs. 15a and 15b. Note that both a low log-loss and a large gap to the best one indicate worse performance. In both scenarios, the performance of **Naive** represented the optimal results since it learned a new structure in every iteration using all the past observations. As one can observe from Figs. 15a and 15b, **MAP** showed a much lower log-loss compared to **Naive**. This is because it was locked to a poor structure initially and failed to infer a proper structure to adapt to the visual context based on limited information obtained from the prior model. In contrast, **Incremental** exhibited closer performance to

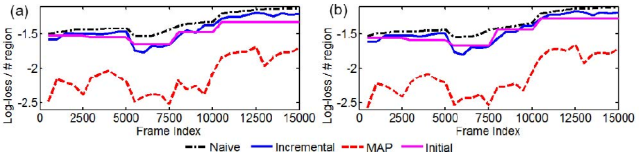


Fig. 15. The log-loss performance yielded by different incremental structure learning methods and a model without incremental learning, in a scenario where (a) Cam 5 was removed starting from frame 7,500 and (b) Cam 5 was added starting from frame 7,500.

Naive by just maintaining a small amount of sufficient statistics. Note that there was a drop of log-loss performance over all methods from frame 5,000 to 7,500 due to a global unusual event due to faulty train (see Fig. 11, for example, frames of this unusual event). Without support from all previously seen observations, **Incremental** exhibited a larger drop as compared to **Naive** during the occurrence of the unusual incident, causing a log-loss gap between the **Incremental** and **Naive** methods. Cam 5 was added/removed right after the end of the unusual incident at frame 7,500. The log-loss gap remained between **Incremental** and **Naive** after frame 7,500 since **Incremental** needed to accumulate new sufficient statistics for the learning of new dependency links when Cam 5 was added/removed from the network. Nevertheless, it quickly approached the distribution modeled by **Naive** thereafter. It is observed from Figs. 15a and 15b that without incremental structure learning (**Initial**), a model was not able to adapt to the current visual context, resulting in relatively lower log-loss performances (also further away from the optimal performance yielded by **Naive**) as compared to **Incremental** after a camera was added/removed from the camera network.

5 Conclusions

We have presented a novel approach to learn time delayed activity dependencies for global unusual event detection in multiple disjoint cameras. Time delayed dependencies are learned globally using a new incremental two-stage structure learning method. Extensive experiments on synthetic data and public scene data have demonstrated that the new approach outperforms methods that disregard the time delay factor or without learning dependency structure globally. Contrary to most existing methods that assume a static model, the proposed approach updates the activity model's parameters and structure incrementally and adaptively to accommodate both gradual and abrupt context changes. There are a number of areas to improve on:

Multimode time delay—Our current method assumes single-mode time delay between regions. There are a number of ways to extend the current method to cope with multimode time delay. One way is to combine the method in [8] with the proposed framework. However, the method proposed in [8] cannot be used directly. This is because although it could take regional activity values as input, feeding the regional activity patterns directly into the method [8] can produce the transformation entropy of activity patterns but not the multimode time delays. In order to obtain the multimode time delays, the input to [8] must be the transition times between departure and arrival observations. To overcome this problem, one could obtain different time delay peaks by performing sliding window-based TDMI analysis on the regional activity patterns. The different time delays can then be fed into [8] for multimodal time distribution modeling. Alternatively, one could examine the TDMI functions for multiple time delay peaks. In the current framework, only the maximum value is considered for each TDMI function. Discovering multiple peaks would naturally lead to the learning of multimodal time delays.

False edge orientations and dependencies—The current method of determining the orientation of edges may lead to false orientations when noise or constant crowdedness are observed in two region pairs. Under these circumstances, spurious peaks may be detected in the TDMI function, which will lead to false edge orientations and dependencies (e.g., 72 → 20 and 82 → 41 in Fig. 6) that do not necessarily correspond to the true time delayed dependencies. A possible way to filter out these false dependencies is by analyzing the shape of the TDMI function. The rationale of carrying out the shape analysis is that a pair of connected regions would typically produce a function with a bell-shaped curve, while a function for two independent regions often exhibits a more random shape with multiple peaks.

Detecting both global and local unusual events—Our method is designed mainly for detecting global unusual events, i.e., context-incoherent patterns across multiple disjoint cameras, rather than local events such as individual suspicious behavior. For the latter, one can consider many existing approaches that are developed specifically for detecting local events [1], [2], [3], [4]. One could achieve both global and local-level detection by combining the proposed method with the existing approaches using different fusion techniques such as score-level fusion or mixture of experts.

Time delayed dependency learning—The Granger causality [41] can be considered to strengthen the current approach by providing a more stringent criterion in reasoning dependency between regional activity patterns.

References

[1] T. Xiang and S. Gong, "Video Behavior Profiling for Anomaly Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 893-908, May 2008.

[2] J. Kim and K. Grauman, "Observe Locally, Infer Globally: A Space-Time MRF for Detecting Abnormal Activities with Incremental Updates," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 2921-2928, 2009.

[3] T. Hospedales, S. Gong, and T. Xiang, "A Markov Clustering Topic Model for Mining Behaviour in Video," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 1165-1172, 2009.

[4] D. Kuettel, M.D. Breitenstein, L.V. Gool, and V. Ferrari, "What's Going On? Discovering Spatio-Temporal Dependencies in Dynamic Scenes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1951-1958, 2010.

[5] T. Xiang and S. Gong, "Incremental and Adaptive Abnormal Behaviour Detection," *Computer Vision and Image Understanding*, vol. 111, no. 1, pp. 59-73, 2008.

[6] E.E. Zelniker, S. Gong, and T. Xiang, "Global Abnormal Behaviour Detection Using a Network of CCTV Cameras," *Proc. IEEE Int'l Workshop Visual Surveillance*, 2008.

[7] D. Makris, T. Ellis, and J. Black, "Bridging the Gaps between Cameras," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 205-210, 2004.

[8] K. Tieu, G. Dalley, and W.E.L. Grimson, "Inference of Non-Overlapping Camera Network Topology by Measuring Statistical Dependence," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 1842-1849, 2005.

[9] O. Javed, K. Shafique, and M. Shah, "Appearance Modeling for Tracking in Multiple Non-Overlapping Cameras," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 26-33, 2005.

[10] X. Wang, K. Tieu, and W.E.L. Grimson, "Correspondence-Free Activity Analysis and Scene Modeling in Multiple Camera Views," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 56-71, Jan. 2010.

[11] H. Zhou and D. Kimber, "Unusual Event Detection via Multi-Camera Video Mining," *Proc. IEEE Int'l Conf. Pattern Recognition*, pp. 1161-1166, 2006.

[12] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*, second ed. The MIT Press, 2000.

[13] G.F. Cooper, "A Simple Constraint-Based Algorithm for Efficiently Mining Observational Databases for Causal Relationships," *Data Mining and Knowledge Discovery*, vol. 1, no. 2, pp. 203-224, 1997.

[14] G.F. Cooper and E. Herskovits, "A Bayesian Method for the Induction of Probabilistic Networks from Data," *Machine Learning,* vol. 9, no. 4, pp. 309-347, 1992.

[15] D. Heckerman, D. Geiger, and D.M. Chickering, "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data," *Machine Learning,* vol. 20, no. 3, pp. 197-243, 1995.

[16] I. Tsamardinos, L.E. Brown, and C.F. Aliferis, "The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm," *Machine Learning,* vol. 65, no. 1, pp. 31-78, 2006.

[17] X. Chen, G. Anantha, and X. Lin, "Improving Bayesian Network Structure Learning with Mutual Information-Based Node Ordering in the K2 Algorithm," *IEEE Trans. Knowledge and Data Eng.,* vol. 20, no. 5, pp. 628-640, May 2008.

[18] N. Friedman and M. Goldszmidt, "Sequential Update of Bayesian Network Structure," *Proc. 13th Conf. Uncertainty in Artificial Intelligence,* pp. 165-174, 1997.

[19] S.H. Nielsen and T.D. Nielsen, "Adapting Bayes Network Structures to Non-Stationary Domains," *Int'l J. Approximate Reasoning,* vol. 49, no. 2, pp. 379-397, 2008.

[20] D.M. Chickering, D. Geiger, and D. Heckerman, "Learning Bayesian Networks: Search Methods and Experimental Results," *Proc. Int'l Workshop Artificial Intelligence and Statistics,* pp. 112-128, 1995.

[21] N. Friedman, I. Nachman, and D. Peér, "Learning Bayesian Network Structure from Massive Datasets: The Sparse Candidate Algorithm," *Proc. 15th Conf. Uncertainty in Artificial Intelligence,* pp. 206-215, 1999.

[22] C.C. Loy, T. Xiang, and S. Gong, "Modelling Activity Global Temporal Dependencies Using Time Delayed Probabilistic Graphical Model," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 120-127, 2009.

[23] C.C. Loy, T. Xiang, and S. Gong, "Time-Delayed Correlation Analysis for Multi-Camera Activity Understanding," *Int'l J. Computer Vision,* vol. 90, no. 1, pp. 106-129, 2010.

[24] L. Zelnik-Manor and P. Perona, "Self-Tuning Spectral Clustering," *Proc. 18th Ann. Conf. Advances in Neural Information Processing Systems,* pp. 1601-1608, 2004.

[25] X. Wang, X. Ma, and W.E.L. Grimson, "Unsupervised Activity Perception in Crowded and Complicated Scenes using Hierarchical Bayesian Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 3, pp. 539-555, Mar. 2009.

[26] Y. Yang, J. Liu, and M. Shah, "Video Scene Understanding Using Multi-Scale Analysis," *Proc. IEEE Int'l Conf. Computer Vision,* 2009.

[27] L. Kratz and K. Nishino, "Anomaly Detection in Extremely Crowded Scenes Using Spatio-Temporal Motion Pattern Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1446-1453, 2009.

[28] A.M. Fraser and H.L. Swinney, "Independent Coordinates for Strange Attractors from Mutual Information," *Physical Rev.,* vol. 33, no. 2, pp. 1134-1140, 1986.

[29] C. Chow and C. Liu, "Approximating Discrete Probability Distributions with Dependence Trees," *IEEE Trans. Information Theory,* vol. 14, no. 3, pp. 462-467, May 1968.

[30] R.C. Prim, "Shortest Connection Networks and Some Generalizations," *Bell System Technical J.,* vol. 36, pp. 1389-1401, 1957.

[31] X. Chen, G. Anantha, and X. Wang, "An Effective Structure Learning Method for Constructing Gene Networks," *Bioinformatics,* vol. 22, no. 11, pp. 1367-1374, 2006.

[32] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms,* third ed. The MIT Press, 2009.

[33] G. Schwarz, "Estimating the Dimension of a Model," *The Annals of Math. Statistics,* vol. 6, no. 2, pp. 461-464, 1978.

[34] N. Friedman and D. Koller, "Being Bayesian about Network Structure," *Proc. 16th Conf. Uncertainty in Artificial Intelligence,* pp. 201-210, 2000.

[35] C.P. de Campos, Z. Zeng, and Q. Ji, "Structure Learning of Bayesian Networks Using Constraints," *Proc. Int'l Conf. Machine Learning,* pp. 113-120, 2009.

[36] C. Auliac, V. Frouin, X. Gidrol, and F. díAlch Buc, "Evolutionary Approaches for the Reverse-Engineering of Gene Regulatory Networks: A Study on a Biologically Realistic Dataset," *BMC Bioinformatics,* vol. 9, pp. 1-14, 2008.

[37] K.P. Murphy, "Active Learning of Causal Bayes Net Structure," technical report, Univ. of California, 2001.

[38] *Markov Chain Monte Carlo in Practice,* W. Gilks, S. Richardson and D. Spiegelhalter, eds., first ed. Chapman and Hall, 1995.

[39] W. Lam and F. Bacchus, "Using New Data to Refine a Bayesian Network," *Proc. 10th Conf. Uncertainty in Artificial Intelligence,* 1994.

[40] D. Eaton and K. Murphy, "Bayesian Structure Learning Using Dynamic Programming and MCMC," *Proc. 23rd Conf. Uncertainty in Artificial Intelligence,* pp. 101-108, 2007.

[41] K. Prabhakar, S. Oh, P. Wang, G.D. Abowd, and J.M. Rehg, "Temporal Causality for the Analysis of Visual Events," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.

Chen Change Loy received the PhD degree in computer science from Queen Mary University of London in 2010. He is now a postdoctoral researcher in the School of Electronic Engineering and Computer Science, Queen Mary University of London. His research interests include computer vision and machine learning, with focus on activity analysis and understanding in surveillance video. He is a member of the IEEE.

Tao Xiang received the PhD degree in electrical and computer engineering from the National University of Singapore in 2002. He is currently a senior lecturer in the School of Electronic Engineering and 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 has published more than 80 papers and coauthored the book *Visual Analysis of Behaviour: From Pixels to Semantics*.

Shaogang Gong received the DPhil in computer vision from Keble College, Oxford University in 1989. He is a professor of visual computation at Queen Mary University of London, a fellow of the Institution of Electrical Engineers, and a fellow of the British Computer Society. He has published more than 250 papers in computer vision and machine learning, the book *Visual Analysis of Behaviour: From Pixels to Semantics*, 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, visual behavior profiling, and recognition.