A Unified Framework for Event Summarization and Rare Event Detection

Junseok Kwon and Kyoung Mu Lee
Department of EECS, ASRI, Seoul National University, 151-742, Seoul, Korea
{paradis0, kyoungmu}@snu.ac.kr, http://cv.snu.ac.kr

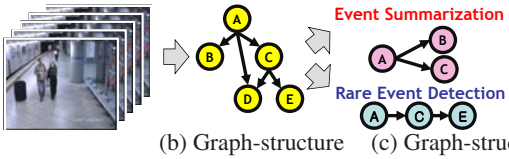
Abstract

A novel approach for event summarization and rare event detection is proposed. Unlike conventional methods that deal with event summarization and rare event detection independently, we solve them together by transforming the problems into a graph editing framework. In our approach, a video is represented as a graph, in which each node of the graph indicates an event obtained by segmenting the video spatially and temporally, while edges between nodes describe the events related to each other. Based on the degree of relations, edges have different weights. After learning the graph structure, our method edits the graph by merging its subgraphs or pruning its edges. The graph is edited toward minimizing a predefined energy model with the Data-Driven Markov Chain Monte Carlo method. The energy model consists of several parameters that represent causality, frequency, and significance of events. We design a specific energy model utilizing these parameters to satisfy each objective of event summarization and rare event detection. Experimental results show that the proposed approach accurately summarizes a video in a fully unsupervised manner. Moreover, the experiments also demonstrate that the approach is advantageous in detecting the rare transition of events.

1. Introduction

Recently, there has been a growing interest in video analysis. Given the large amount of video data, the key objective of video analysis is to analyze the data automatically and then extract useful information from it efficiently. Among the various problems of video analysis, *event summarization* and *rare event detection* are being addressed by a growing number of researchers owing to the increasing interest on intelligent surveillance systems [3, 7, 17]. The aim of this paper is to develop a fully automatic system that can solve these two problems efficiently and robustly in a single framework.

The goal of event summarization is to condense a long video into a short one by extracting the story-line of the



(a) Input video (b) Graph-structure learning (c) Graph-structure editing

Figure 1. **Overview of our system** The system consists of two parts. The first part is graph learning for analyzing video data. The second is graph editing for extracting useful information (event summarization and rare event detection) from the analyzed data.

video [2, 4, 6, 10, 9, 15, 16]. The story-line consists of representative events of the video, which are rearranged according to causality between events. Gupta et al. [4] accurately extracted visually grounded story-line learned from annotated videos. However, the method has weaknesses such as the need for labeled data to learn the story-line. On the other hand, our method can extract the story-line in a fully unsupervised manner. Hospedales et al. [6] and Kuettel et al. [9] introduced a Markov Clustering Topic Model and a Dependent Dirichlet Processes-Hidden Markov Model, respectively, and successfully detected interesting events and their relations in complex and crowded public scenes. Compared to these methods, our method can find more complex relations of events by measuring the causality, frequency, or significance of events.

In rare event detection, unusual events are detected automatically [1, 8, 11, 18, 19, 20, 23, 24]. It is necessary for surveillance systems since such events should be reported for further examination. Xiang et al. [20] suggested a surveillance system for recognizing normal behavior in real-time and detecting abnormal actions simultaneously. Boiman et al. [1] detected irregular behavior by comparing visual data with the database containing regular patterns. However, these two algorithms cannot detect unusual transitions between events, such as an unusual switch in a series of actions. In contrast, the proposed algorithm accurately finds these transitions.

The philosophy of our method is that an input video can be represented as a single graph and, by editing this graph, the problems of event summarization and rare event detection can be efficiently solved. Fig.1 summarizes the overall procedure of our system. The system first transforms an in-

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

(a) Spatial Decomposition (b) Temporal Decomposition (c) Generated nodes

Figure 2. **Process of learning nodes** Our system decomposes the video spatially and temporally into three-dimensional segments.

put video into a graph as shown in Fig.1(b). In the graph, the nodes indicate events contained in the video, which are obtained by segmenting the video spatially and temporally. The edges are connected when relations between events exist. The system finds the relations utilizing the data-mining technique in [5]. Depending on the process of editing the graph, different problems can be solved as illustrated in Fig.1(c). In the event summarization problem, the graph is edited to leave events with high causality. On the other hand, rare events are detected by leaving events with high causality but low frequency. We define rare events as those with low-frequency since the definition is widely used in the rare event detection problems [20, 24]. In all problems, events with low significance are deleted because the estimated causality and frequency of the events are not reliable in these cases.

The first contribution of this paper is to present the completely unsupervised method for extracting the story-line and detecting rare events in the video (*automatic*). The second contribution is the capacity to solve the problems of event summarization and rare event detection in a single framework by considering them as a graph learning and editing one (*efficiency*). The last contribution is the ability of the proposed method to discover associated events from a video (*usefulness*). With these contributions, our system satisfies the aforementioned key objective of video analysis, which is to analyze data *automatically* and extract *useful* information from it *efficiently*.

2. Video-Structure Graph Learning

In this section, the process of learning the nodes (subsection 2.1) and edges (subsection 2.2) of a graph from an input video is explained.

2.1. Learning Nodes

A node represents a spatio-temporal event. Then, the nodes of the graph are obtained by decomposing a video into three-dimensional segments, where each segment corresponds to each node of the graph. To this end, we adopt the method introduced in [13] and extend it by decomposing the video not only spatially but also temporally. [1] Fig.2 describes the whole process of learning the nodes, which mainly consists of two steps, spatial decomposition and temporal decomposition.

***Spatial Decomposition*:** The first step is to decompose a video into several regions in space as illustrated in Fig.2(a) because a different event occurs depending on the spatial position of a region. In the subway scene, the region of the subway rail is where the event of arrival and departure of a train occurs. On the other hand, people typically get in and out of the train at the region of the subway platform. The system automatically finds the boundaries of these semantic regions by dividing the video into blocks of 10×10 pixels and clustering the blocks according to the similarity of local spatio-temporal activity patterns. With regard to the features describing the activity patterns, we utilize the percentage of static foreground pixels within the block and the percentage of pixels within the block that are classified as moving foreground [13]. Static foreground pixels are naively detected by subtracting each frame with a background image while moving foreground pixels are found by subtracting consecutive frames. As to the clustering method, the spectral clustering algorithm in [21] is employed, where the number of regions is determined automatically. Note that the spatial decomposition is done by considering all frames in the video.

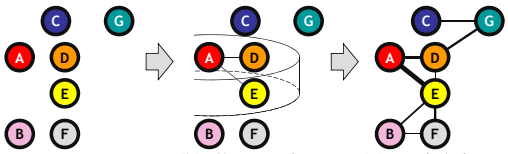
***Temporal Decomposition*:** After decomposing the video into several regions spatially, each region is further divided temporally as shown in Fig.2(b) since different events may occur in relation to time even at the same region. For example, in the subway scene, two different events occur at the same region of the subway rail: the arrival and the departure of a subway train. The system separates the event of each region into multiple events in a manner similar to that in spatial decomposition. However, features describing the activity patterns are different from those in the spatial decomposition case. For the feature, the dominant magnitude and angle of optical flow are utilized, where optical flow is calculated by the Lucas-Kanade method in [14]. The dominant magnitude of optical flow represents the average movement of events. The dominant angle of optical flow indicates the representative direction of events.

Note that our method does not need the correspondence across nodes. If the events are similar each other, they are clustered in a node during the process of spectral clustering using [13]. So, each node includes totally different events. For instance, all events of the train approaching are clustered in node A and all events of the train departure are clustered as node B. In this case, an example of the constructed graph is $A \rightarrow B \rightarrow A$ according to the temporal order of the video.

2.2. Learning Edges

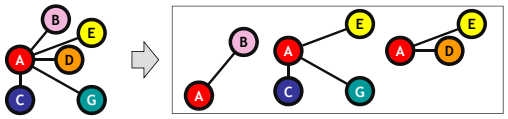
To construct edges in the graph, each node is connected with its neighbors. The edges are then weighted according to properties such as causality, frequency, and signif-

[1] If we employ more advanced video segmentation and image representation methods [22], our method can further improve the performance.



(a) Generated nodes (b) Connecting edges (c) Weighting edges (initial graph)

Figure 3. **Process of learning edges** Our system connects neighbor nodes while weighting on the edges according to causality, frequency, and significance of events. Then, (c) is an initial graph for the graph-structure editing in the next section.



(a) Neighbors of node A (b) Example of hypotheses for node A

Figure 4. **Process of making hypotheses of relations**

icance of events. We estimate the properties utilizing the data-mining technique introduced in [5]. Fig.3 describes the whole process of generating edges.

***Connecting edges*:** Each node of the graph is connected to its neighboring nodes. This implies that, if events occur at a closer location, then there is a higher probability that these events are related to each other. To determine the neighbors of each node, we draw an imaginary cylinder centered at each node. If a certain node falls in the cylinder centered at node A, the node is considered to be a neighbor of node A, as illustrated in Fig.3(b). Then, the initial graph certainly takes the temporal order of two nodes into account This is because the nodes of the initial graph are connected with edge only when they are neighbors spatially and temporally. Note that the width (height) of a cylinder indicates how far a node is considered to be a neighbor at the axis of space (time). In the experiments, width and height of the cylinder are set at half of the diagonal length of a scene and at 100 frames, respectively.

***Making hypotheses*:** After connecting the edges, the system makes multiple hypotheses of the relations between nodes. To this end, we select a node in the graph as the center-node and consider the subgraph for the center-node, which consists of neighbor nodes and edges connected to it as shown in Fig.4(a). In the subgraph, multiple hypotheses are created by deleting a different subset of edges and nodes from the subgraph, as illustrated in Fig.4(b). Our method makes all possible hypotheses for each center-node. Finally, the hypothesis is duplicated in proportion to the temporal length of the center-node. If there are p number of neighbors and the number of frames in the center-node is q, the total number of hypotheses is calculated by $q\sum_{i=1}^{p}\begin{pmatrix} p \\ i \end{pmatrix}$. This process is performed repeatedly until all nodes in the graph are selected as the center-node. Fig.4 illustrates an example of hypotheses when node A is selected as center-node. In this example, the total number of hypotheses is $q\sum_{i=1}^{5}\begin{pmatrix} 5 \\ i \end{pmatrix} = 29q$.

From the hypotheses of the relations between nodes, our system derives statistics such as $m(A,B)$, $m(A)$, n_p, and n_e where $m(A,B)$ is the observed frequency of events A and B occurring jointly, $m(A)$ is the observed frequency when event A occurs, n_p is the number of all pairs of events, and n_e is the number of events in the hypotheses. Note that our method can obtain the statistics using only one video. The statistic of $m(A,B)$ can be obtained by counting the number of hypotheses, which include event A and event B simultaneously. For example, if the constructed graph is $A \rightarrow B \rightarrow A$, possible hypotheses are $A \rightarrow B$ and $B \rightarrow A$. In this case, $m(A,B)$ is 2. In the London traffic sequence, video length was one hour and $m(gostraight, turnleft)$ was 1497. n_p and n_e denote the number of all possible pairs and events, shown in the hypotheses, respectively.

***Weighting edges*:** Events connected with edges imply that they have a certain type of a relation. Characteristics of relations are distinguished by utilizing the aforementioned statistics obtained from the hypothesis. The process of finding characteristics is called weighting edges because the strength of the relations is determined by the characteristics. In our problem, three different types of characteristics are used, namely, causality, frequency, and significance of events.

The causality $c(A \rightarrow B)$ represents the probability that event B is caused by event A:

$$c(A \rightarrow B) = \frac{p(A,B)}{p(A)} = \frac{n_e m(A,B)}{n_p m(A)}. \quad (1)$$

If related events have a high value on the causality, the system considers them to have a strong causal relationship. On the other hand, the frequency $f(A \rightarrow B)$ indicates the probability that events A and B occur jointly:

$$f(A \rightarrow B) = p(A,B) = \frac{m(A,B)}{n_p}. \quad (2)$$

If related events have a low value on the frequency, then the relation is considered abnormal. The last characteristic is the significance of events called the p-value, which measures how much events A and B are independent:

$$p = \sum_{i=m(A,B)}^{m(A)} \begin{pmatrix} n_p \\ i \end{pmatrix} (p(A)p(B))^i (1 - p(A)p(B))^{n_p - i}. \quad (3)$$

The p-value can be used to measure the confidence of causality and frequency of events. If the p-value is very high, events are highly independent. In this case, the confidence of causality and frequency is very low because the independence of events offer no sufficient chances to obtain reliable values of causality and frequency.

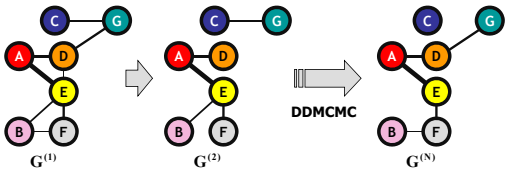


Figure 5. **Process of editing the graph-structure** Utilizing DDMCMC, our method obtains N samples of the graph-structures. Among the samples, the method selects the best graph-structure as the final result of event summarization or rare event detection.

In practice, the binomial probability in (3) is quite difficult to calculate. Thus, we obtain the z-score [5] which is an upper bound of (3) instead of directly calculating p-value:

$$z(A \rightarrow B) = \frac{m(A,B) - n_p p(A)p(B)}{\sqrt{n_p p(A)p(B)(1-p(A)p(B))}}, \quad (4)$$

where $z(A, B)$ denotes the z-score of events A and B.

3. Video-Structure Graph Editing

After learning the initial graph such like Fig.3(c), our system edits the graph toward minimizing a predefined energy model with the Data-Driven Markov Chain Monte Carlo (DDMCMC)[2] method as described in Fig.5. The energy model is composed of several variables representing causality, frequency, and significance of events and designed with these variables to satisfy each objective of the event summarization and rare event detection problem.

3.1. Energy Minimization by DDMCMC

The best graph $\hat{\mathbf{G}}$ is one that gives the minimum log-likelihood estimate over the N number of samples.

$$\hat{\mathbf{G}} = \underset{\mathbf{G}^{(l)}}{\arg min} - \log p(\mathbf{Y}|\mathbf{G}^{(l)}) \quad for\ l = 1, \ldots, N, \quad (5)$$

where $\mathbf{G}^{(l)}$ indicates the l-th sample of the graph-structure; Y denotes the observations of $\mathbf{G}^{(l)}$, which are causality, frequency, and significance of events in $\mathbf{G}^{(l)}$; and $-\log p(\mathbf{Y}|\mathbf{G}^{l})$ represents the energy model that measures how much $\mathbf{G}^{(l)}$ and optimal graph-structure $\mathbf{G}^{opt}$ coincide. In (5), the DDMCMC method can be interpreted as a data-driven stochastic search, where $\hat{\mathbf{G}}$ may tend to a structure resembling $\mathbf{G}^{opt}$.

The DDMCMC method consists of two main steps: the proposal step and the acceptance step. In the proposal step, a new graph-structure is proposed by the proposal density function. Given the proposed graph-structure, the method decides whether it is accepted or not with the acceptance ratio in the acceptance step:

$$\gamma = min\left[1, \frac{(-\log p(\mathbf{Y}|\mathbf{G}^*))^{-1} Q(\mathbf{G};\mathbf{G}^*)}{(-\log p(\mathbf{Y}|\mathbf{G}))^{-1} Q(\mathbf{G}^*;\mathbf{G})}\right], \quad (6)$$

[2]The MCMC method is called as Data-driven one [12] because the proposal distributions in (8), (9), (11), and (12) are strongly guided by the observation data such as the causality and frequency.

Algorithm 1 Graph-structure editing

```
for l=1 to N − 1 do
    G = G^(l)
    Propose G* using Q(G*; G) in (7).
    Calculate γ in (6) with − log p(Y|G) of (10)(13).
    θ ∼ U[0, 1].
    if θ < γ then
        G^(l+1) = G*
    else
        G^(l+1) = G
    end if
end for
```

where $Q(\mathbf{G}^*;\mathbf{G})$ denotes the proposal density function and $\mathbf{G}^*$ represents the new graph-structure proposed by $Q(\mathbf{G}^*;\mathbf{G})$. These two steps iteratively go on until the number of iterations reaches a predefined value, as summarized in Algorithm 1.

To calculate the acceptance ratio in (6), the remaining task is to design the proposal density function $Q(\mathbf{G}^*;\mathbf{G})$ and the energy model $-\log p(\mathbf{Y}|\mathbf{G})$. In the next subsection, it is explained how to efficiently design the proposal density function for increasing the accuracy of the estimate in (5) given a fixed number of samples and build the energy model appropriately to satisfy each objective of event summarization and rare event detection.

3.2. Event Summarization

Our goal of the event summarization problem is to extract representative and interesting events having strong causality to each other from a video. To achieve this, the proposal density function in (6) is designed as follows:

$$Q(\mathbf{G}^*;\mathbf{G}) = \begin{cases} Q_1(\mathbf{G}^*;\mathbf{G}) : \text{with the probability } \frac{1}{2} \\ Q_2(\mathbf{G}^*;\mathbf{G}) : \text{with the probability } \frac{1}{2}, \end{cases} \quad (7)$$

where $Q_1(\mathbf{G}^*;\mathbf{G})$ adds a new pair of events into the current graph, G, and $Q_2(\mathbf{G}^*;\mathbf{G})$ deletes an existent pair of events from G. In $Q_1(\mathbf{G}^*;\mathbf{G})$, the candidate pair of events, A and B, is chosen for addition into the graph with the probability:

$$p_a(A \rightarrow B) = \frac{exp^{-(1-c(A\rightarrow B))}}{\sum_{\forall r\rightarrow s \in R'} exp^{-(1-c(r\rightarrow s))}}, \quad A \rightarrow B \in R', \quad (8)$$

where $c(A \rightarrow B)$ represents the causality between events A and B calculated by (1) and R' denotes the set of all pairs of related events, which are not included in the current graph, G. In $Q_2(\mathbf{G}^*;\mathbf{G})$, the existent pair of events, A and B, is chosen for deletion from the graph with the probability:

$$p_d(A \rightarrow B) = \frac{exp^{-c(A\rightarrow B)}}{\sum_{\forall r\rightarrow s \in R} exp^{-c(r\rightarrow s)}}, \quad A \rightarrow B \in R, \quad (9)$$

where R denotes the set of all pairs of related events existing in G.

The energy model of the event summarization problem is designed to include as many related events as possible while the events have high values on causality and significance.

$$
\begin{aligned}
&-\log p(\mathbf{Y}|\mathbf{G}) = -\log p(c, z|\mathbf{G}) \\
&= \lambda_c \sum_{\forall r \to s \in R} (1 - c(r \to s)) \\
&+ \lambda_s \sum_{\forall r \to s \in R} z(r \to s) - \lambda_n |G|^2,
\end{aligned} \tag{10}
$$

where $c(r \to s)$ and $z(r \to s)$ represent the causality and significance of a pair of events calculated by (1) and (4), respectively; λ_c, λ_s and λ_n indicate the weighting parameters; and $|G|$ is the regularization or prior term, which returns the total number of events in graph G. Note that our system has the ability to control the length of the video's story-line by differently weighting on $|G|$ with the weighting parameter λ_n in (10). The system with the lower value λ_n obtains a more concise story-line of the video.

3.3. Rare Event Detection

In the rare event detection problem, our system finds events that have high causality but low frequency. To this end, the proposal density function is the same with (7) except in choosing a candidate pair of events for adding or deleting. In rare event detection, the candidate pair of events, A and B, is chosen for addition to the graph with the probability:

$$
p_a(A \to B) = \frac{exp^{-(1-c(A\to B))-f(A\to B)}}{\sum_{\forall r \to s \in R'} exp^{-(1-c(r\to s))-f(r\to s)}}, \tag{11}
$$

where $A \to B \in R'$ and $f(A \to B)$ represents the frequency of the pair of events A and B, calculated by (2). Similarly, the existent pair of events, A and B, is chosen for deletion from the graph with the probability:

$$
p_d(A \to B) = \frac{exp^{-c(A\to B)-(1-f(A\to B))}}{\sum_{\forall r \to s \in R} exp^{-c(r\to s)-(1-f(r\to s))}}, \tag{12}
$$

where $A \to B \in R$.

Our system designs the energy model of rare event detection toward including related events with high values on causality and significance but low values on frequency while maintaining a certain number of events.

$$
\begin{aligned}
&-\log p(\mathbf{Y}|\mathbf{G}) = -\log p(c, f, z|\mathbf{G}) \\
&= \lambda_c \sum_{\forall r \to s \in R} (1 - c(r \to s)) + \lambda_f \sum_{\forall r \to s \in R} f(r \to s) \\
&+ \lambda_s \sum_{\forall r \to s \in R} z(r \to s) - \lambda_n |G|^2,
\end{aligned} \tag{13}
$$

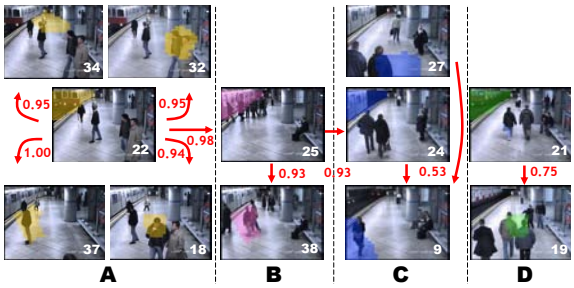


Figure 6. **Results of event summarization recovered by our method** in the subway platform sequence. The number in each rectangle denotes a node index. The red arrows represent the existence of significant relations between nodes while the red numbers indicate causality.

λ_c, λ_s, λ_f, and λ_n indicate the weighting parameters. Note that the rare event detection problem includes event summarization in our framework. If λ_f in (13) is taken as zero, the equations are exactly the same with (10) of the event summarization problem, respectively.

4. Experimental Results

In the experiment, we tested three datasets which are publicly available [3]. Using the datasets, the proposed algorithm [4] was compared with the significant association rule model (SARM) in [5] and Dependent Dirichlet Processes-Hidden Markov Model (DDP-HMM) in [9], which are the *state-of-the-art* data mining and event detection algorithm, respectively. For the SARM, the same graph learned by our method was utilized as the initial graph. Then, utilizing the data mining techniques in [5], the graph was edited to produce event summarization and rare event detection results. For the DDP-HMM, we used the software provided by authors. We adjusted parameters of SARM and DDP-HMM to show the best performance.

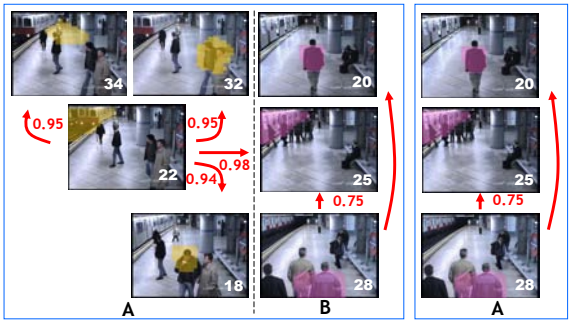
4.1. The Subway Platform Sequence

For event summarization, the energy model in (10) was minimized with the DDMCMC method where the number of used samples was 800 and λ_c, λ_s, and λ_n were set to 100, 0.01, and 1, respectively. After energy minimization, the edited graph was obtained as illustrated in Fig.6. The graph consists of 12 nodes and 10 edges while the original graph has 45 nodes and 231 edges. The extracted story-line of the video can be divided into the following four parts.

• **Part A :** This part describes the approach of a train. When the train pulls into the platform (node 22), waiting passengers typically converge near the train (node 34, 37), although a small number of people move toward the down direction of the scene (node 18, 32). Our system summarized both normal (node 34, 37) and abnormal (node 18,

[3]The details of the datasets can be found at following sites. i-LIDS (http: //www.eecs.qmul.ac.uk/ ∼andrea/avss2007_d.html), BOSS (http: //193.190.194.199/BOSS), London Traffic from [6]

[4]The result videos can be found at http://cv.snu.ac.kr/.



(a) $\lambda_n = 0.5$ (b) $\lambda_n = 0.3$

Figure 7. **More concise results of event summarization** compared to Fig.6 ($\lambda_n = 1$) in the subway platform sequence.

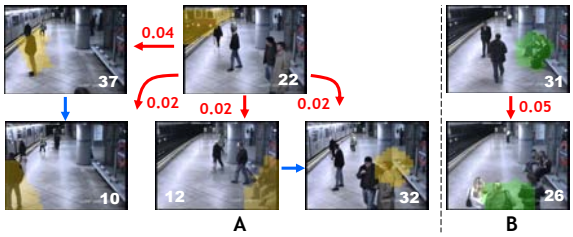


Figure 8. **Results of rare event detection found by our method** in the subway platform sequence. The red arrows represent the existence of significant relations between nodes with low frequency, denoted by red numbers. The blue arrows describe that the nodes are spatial or temporal neighbors to each other.

32) events that occurred at the platform well, which were caused by the event of the approaching train (node 22).

• **Part B :** The second part shows related events after the train opens its doors. When the doors of the train are opened (node 25), people get on or off the train (node 38). Although there are severe occlusions between people, the system captured events, which have casual relations, robustly.

• **Part C :** The third part concerns the events that arise from the departure of the train from the subway station. After the train leaves the station (node 24)), some people move to the top of the platform (node 27), while others, who get off the train, move to the bottom of the platform (node 9).

• **Part D :** The last part illustrates the situation when a train approaches the platform from the opposite side (node 21). While the event is occurring, people go to the upside or downside of the platform (node 19). The events included in part D are very difficult to extract since they are severely rare events in the subway platform sequence.

While the story-line in Fig.6 is the long version of the event summarization, the story-lines in Fig.7 are the concise versions obtained by decreasing the λ_n value in (10). Fig.7(a) illustrates the result of event summarization when λ_n is 0.5. It includes a smaller number of, but more representative, events such as the train approaching the platform and the people getting on or off the train after it opens its doors. Fig.7(b) is the most concise version of the event summarization. It contains most representative events such as people getting on or off the train.

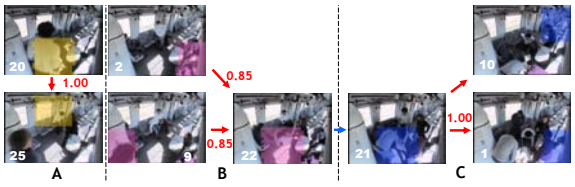
Representative events	Ours	SARM	DDP-HMM
Approaching of the train	**1**	**1**	**1**
Opening train's doors	**1**	0	0
Leaving of the train	**1**	0	0
Approaching (opposite side)	**1**	0	**1**
Waiting the train	2	3	**1**
Getting on or off the train	**3**	6	5
Counterflow	3	**2**	0
Rare transitions	**Ours**	**SARM**	**DDP-HMM**
Interesting	**5**	2	0
Uninteresting	**0**	3	0

Table 1. **Comparison** in the subway platform sequence. Red indicates best performance. If the number is 0, it means that the method could not find the representative event type. On the other hand, number 1 indicates that the method successfully found the event type. If the number is more than 2, it means that the method unnecessarily found the same event type again. In the case of rare event detection, the lager number of rare transitions indicates better results.

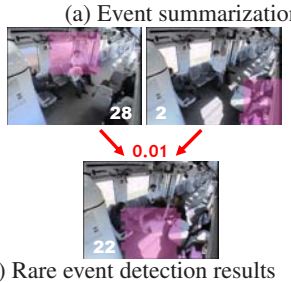
Figure 9. **Selection of representative activities detected by DDP-HMM** in the subway platform sequence.

For rare event detection, λ_c, λ_s, λ_f, and λ_n were set in (13) as 10, 0.01, 100, and 0.5, respectively. After energy minimization of (13), our system obtained two abnormal scenarios as described in Fig.8. In the first scenario, two people who are sitting on the stool (node 12) and a person at the platform (node 37) start to move toward the down direction of the scene (node 10, 32) as the train approaches the platform (node 22). The events of nodes 10 and 32 themselves are actually normal since these types of events frequently occur when people get on or off the train. However, the events become abnormal when their relations with the event of node 22 are considered. The event of node 22 infrequently causes the event of node 10 or 32. Instead, it typically causes most people in the scene go near the approaching train. Thus, it is the *rare transition* of events from node 22 to node 10 or 32. Similarly, the second scenario includes the abnormal scenario where a woman continues to sit on the stool (node 31) although the train had already arrived at the platform. She finally goes out of the subway station with the people who get off the train (node 26) after the train departs from the station.

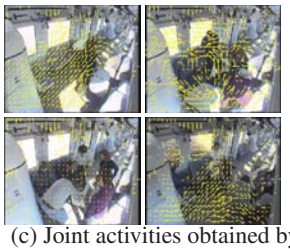
Table 1 shows the comparison with different event summarization and rare event detection methods. We evaluated event summarization by counting the number of representative event types discovered by the methods. We utilize the method in [6] for the evaluation. For this, we manually label each segment in the video for making the ground truth. As shown in Table 1, our method more accurately extracted representative event types and found more interesting and



(a) Event summarization results of our method



(b) Rare event detection results of our method



(c) Joint activities obtained by DDP-HMM

Figure 10. **Results of event summarization and rare event detection made by our method** in the disease sequence. The red numbers represent the causality of events in (a) and frequency of events in (b), respectively.

larger number of events in the sequence compared to SARM and DDP-HMM. Although DDP-HMM also showed good performance by finding diverse moving directions of people that get on or off the train, it missed a few representative events such as "opening train's doors" and "leaving of the train", as shown in Fig. 9. Note that, since DDP-HMM learns the number of activities automatically, we could not increase the total number of representative events to be same as ours. On the other hand, our method produced better results since it uses the foreground percentage as well as the optical flow for the feature and exploits rich statistics about the relation of events such as causality, frequency, or significance of events. Similarly, we evaluated rare event detection by counting the number of rare transitions discovered by the methods. Interesting rare transitions found by our method are people not going near the approaching train when the train was approaching. Our method detected a larger number of rare transitions compared to SARM as described in Table 1. DDP-HMM cannot find rare transitions of events during the training phase, although it can be used to detect them after training a model off-line. On the other hand, given a video, our method needs no training phase to detect rare transitions of events in the video.

4.2. The Disease Sequence

The disease sequence of BOSS dataset was also tested. The sequence includes the scenario where a man enters the railway coach. After a while, he suddenly falls ill. Passengers help him to his seat. Fig. 10(a) and 10(b) show the qualitative results of event summarization and rare event detection in this sequence. For event summarization, λ_c, λ_s, and λ_n in (10) were set as 100, 0.01, and 0.3, respectively. As the energy calculated by (10) is minimized, the graph was edited to contain 8 nodes and 6 edges from the original

Representative events	Ours	SARM	DDP-HMM
Entering the coach	**2**	4	**2**
Taking a sit	2	3	**1**
Falling ill	**1**	0	0
Coming up to him	**2**	0	3
Helping him to his seat	**1**	**1**	0
Rare transitions	**Ours**	**SARM**	**DDP-HMM**
Interesting	**2**	1	0
Uninteresting	**0**	1	0

Table 2. **Comparison** in the disease platform sequence. Red indicates best performance.

31 nodes and 142 edges. As illustrated in Fig. 10(a), our system well summarized the disease sequence with three different parts. The first part describes the scenario where people enter the railway coach (node 20) and take their seats (node 25). The event of node 20 caused the event of node 25. In the second part, our method captured the meaningful situation of the man suddenly falling ill (node 22). The method detected rare transitions between nodes 2 and 9 and node 22. These transitions of events are irregular because the abnormal event of node 22 occurs just after the normal events of nodes 2 and 9. After the man falls ill, passengers help him to his seat (node 21). The event of node 21 causes passengers to rise from their seats to help him (node 1,10). Our method accurately found the casual relationship between node 21 and nodes 1 and 10 as shown in the third part of Fig. 10(a), in spite of severe occlusions and interactions of people. On the other hand, the joint activities obtained by DDP-HMM described inaccurate transitions of events due to the errors occurred by severe occlusion and background clutter, although a full comparison is not possible, as illustrated in Fig. 10(c).

For rare event detection, we set λ_c, λ_s, λ_f, and λ_n in (13) as 10, 0.01, 100, and 0.01, respectively. Fig. 10(b) shows the edited graph obtained by our method for rare event detection. Our system successfully detected the meaningful but abnormal event of the man falling ill (node 22). Additionally, the system found related events with node 22. For example, node 28 includes the region where the man has felt pain for a while. This caused him to fall ill, which is the event of node 22.

As shown in Table 2, our method outperforms SARM and DDP-HMM quantitatively in the performance of both event summarization and rare event detection.

4.3. The London Traffic Sequence

This sequence contains the traffic at intersections. To summarize the sequence, we set λ_c, λ_s, and λ_n in (10) as 100, 0.05, and 0.01, respectively, and minimize the energy of (10). Then, we obtained the edited graph which consists of 14 nodes and 14 edges, while the original one 827 nodes and 2353 edges. As illustrated in Fig. 11, our method summarized events similar to those shown in [6, 9]. It found the

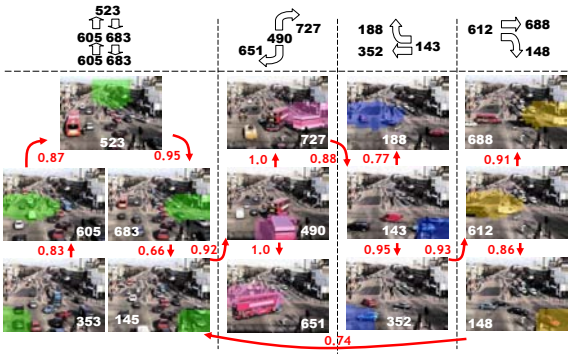
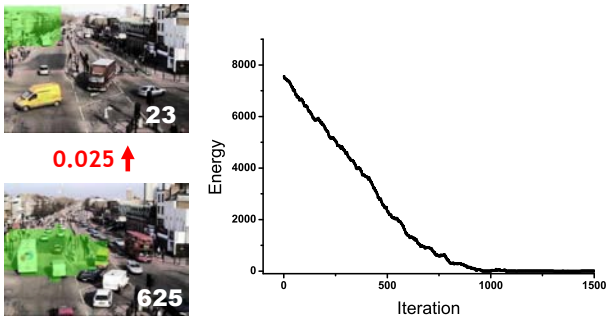


Figure 11. **Results of event summarization recovered by our method** in the traffic sequence. The figures at the top are a simplified version of the figures at the bottom.



(a) Rare event detection results (b) Energy minimization

Figure 12. **Results of rare event detection found by our method** in the traffic sequence.

four representative movements of cars according to traffic lights and recovered the traffic light cycle accurately.

Our method also found a rare transition of events at the same time as illustrated in Fig. 12(a). In the dataset, a car suddenly stopped on the road while other cars usually keep going. It is an unusual switch in a series of car movements, of which frequency is only 0.025. This can not be recovered by DDP-HMM at its training phase, although it may detect the transition after training. On the other hand, our method found both rare and representative sequences of events simultaneously by casting the problems as the graph editing framework. During the process of the method, the energy in (13) was decreased from 7902 to 353 and finally converged as shown in Fig. 12(b), in which λ_c, λ_s, λ_f, and λ_n were set to 100, 0.005, 200, and 0.005, respectively.

Our method spends most computational time to segment the video spatially and temporally by the method in [13]. Thus, the computational cost and scalability highly depend on [13]. By properly optimizing the process, we can greatly enhance the performance although it approximately takes 1 seconds per frame at the current state.

5. Conclusion

In this paper, we have proposed an unified framework for event summarization and rare event detection and presented the graph-structure learning and editing method to solve these problems efficiently. The experimental results demonstrated that the proposed method outperformed conventional algorithms in complex and crowded public scenes by exploiting and utilizing causality, frequency, and significance of relations of events.

Acknowledgement

This research was supported in part by the MKE, Korea and Microsoft Research Asia, under IT/SW Creative research program supervised by the NIPA (NIPA-2011-C1810-1102-0046).

References

[1] O. Boiman and M. Irani. Detecting irregularities in images and in video. *IJCV*, 74(1):17–31, 2007. 1
[2] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. *ICCV*, 2011. 1
[3] R. Collins, A. Lipton, and T. Kanade. Introduction to the special section on video surveillance. *PAMI*, 22(8):745–746., 2000. 1
[4] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. *CVPR*, 2009. 1
[5] W. Hamalainen and M. Nykanen. Efficient discovery of statistically significant association rules. *ICDM*, 2008. 2, 3, 4, 5
[6] T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behaviour in video. *ICCV*, 2009. 1, 5, 6, 7
[7] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. A system for learning statistical motion patterns. *PAMI*, 28(9):1450–1464., 2006. 1
[8] L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. *CVPR*, 2009. 1
[9] D. Kuettel, M. D. Breitenstein, L. V. Gool, and V. Ferrari. Whats going on? discovering spatio-temporal dependencies in dynamic scenes. *CVPR*, 2010. 1, 5, 7
[10] S. Kwak, B. Han, and J. H. Han. Scenario-based video event recognition by constraint flow. *CVPR*, 2011. 1
[11] J. Kwon and K. M. Lee. Simultaneous video synchronization and rare event detection via cross-entropy monte carlo optimization. *Computer Vision Workshops (ICCV Workshops)*, 2009. 1
[12] J. Kwon and K. M. Lee. Tracking by sampling trackers. *ICCV*, 2011. 4
[13] C. C. Loy, T. Xiang, and S. Gong. Multi-camera activity correlation analysis. *CVPR*, 2009. 2, 8
[14] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *IJCAI*, 1981. 2
[15] K. Prabhakar, S. Oh, P. Wang, G. D. Abowd, and J. M. Rehg. Temporal causality for the analysis of visual events. *CVPR*, 2010. 1
[16] Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg. Webcam synopsis: Peeking around the world. *ICCV*, 2007. 1
[17] M. Shah, O. Javed, and K. Shafique. Automated visual surveillance in realistic scenarios. *MultiMedia*, 14(1):30–39, 2007. 1
[18] X. Wang, X. Ma, and E. Grimson. Unsupervised activity perception by hierarchical bayesian models. *PAMI*, 31(3):539–555., 2009. 1
[19] S. Wu, B. E. Moore, and M. Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. *CVPR*, 2010. 1
[20] T. Xiang and S. Gong. Video behavior profiling for anomaly detection. *PAMI*, 30(5):893–908, 2008. 1, 2
[21] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. *NIPS*, 2004. 2
[22] H. Zhang, J. E. Frittsb, and S. A. Goldmana. Detecting irregularities in images and in video. *CVIU*, 110(2):260–280, 2008. 2
[23] B. Zhao, L. Fei-Fei, and E. P. Xing. Online detection of unusual events in videos via dynamic sparse coding. *CVPR*, 2011. 1
[24] H. Zhong, M. Visontai, and J. Shi. Detecting unusual activity in video. *CVPR*, 2004. 1, 2