Semantic Video-to-Video Search using Sub-Graph Grouping and Matching

Tae Eun Choe, Hongli Deng
ObjectVideo
Reston, VA, USA
{tchoe, hdeng}@objectvideo.com

Feng Guo
Walmart Labs
CA, USA
fguo2004@gmail.com

Mun Wai Lee
Intelligent
Automation Inc.
Rockville, MD, USA
mlee@i-a-i.com

Niels Haering
Affectiva, Waltham
MA, USA
haering@gmailcom

Abstract

We propose a novel video event retrieval algorithm given a video query containing grouped events from large scale video database. Rather than looking for similar scenes using visual features as conventional image retrieval algorithms do, we search for the similar semantic events (e.g. finding a video such that a person parks a vehicle and meets with other person and exchanges a bag). Videos are analyzed semantically and represented by a graphical structure. Now the problem is to match the graph with other graphs of events in the database. Since the query video may include noisy activities or some event may not be detected by the semantic video analyzer, exact graph matching does not always work. For efficient and effective solution, we introduce a novel subgraph indexing and matching scheme. Subgraphs are grouped and their importance is further learned over video by topic learning algorithms. After grouping and indexing subgraphs, the complex graph matching problem becomes simple vector comparison in reduced dimension. The performances are extensively evaluated and compared with each approach.

1. Introduction

Given a video as a query, finding semantically closest videos is an emerging research area as large-scale video data are generated and stored. The objective of this study is to retrieve videos containing similar complex activities with the query video rather than finding visually similar videos. Challenges in this work are; (1) to retrieve relevant data efficiently in a very large scale of video data; (2) to be robust to video noises (*e.g.* scale, occlusion, and view-point changes) and systematic noises from not-so-perfect state-of-art object detection and tracking methods; and (3) to model any possible complex events even with limited number of semantic expressions of video events.

Activities in a scene are classified into four categories based on complexity, (1) basic action, (2) action, (3) event, and (4) grouped event. Basic action involves a single agent with simple activities or gestures (e.g. *walk, run, stop, turn, sit, bend, lift hands, etc.*). The action is a single agent interacting with a single subject (e.g. *carry a box, open door, disembark a car, etc.*). The event is defined as a single or multiple agents interacting with a single or multiple subjects (e.g. *Person_1 passes a ball to Person_2.*). The grouped event consists of the two or more events occurring concurrently or sequentially (*e.g. Person_1 disembarks Vehicle_2, meets Person_3, takes a bag_4 from Person_3, and then Person_3 walks away and Person_1 rides Vehicle_2 and leaves the scene*).

For a conventional video retrieval system, color (histogram or correlogram) and visual features (e.g. HOG, SIFT) are commonly used to find similar scenes [3][12][19] rather than activities. Especially in surveillance videos, since the activities are taken at the same sites, conventional retrieval methods cannot detect activities of interest. Meanwhile, Snoek *et al.* retrieve video events using time interval [18] and also propose video retrieval concept detectors which handle multi-modal queries and fuse them to find the best matching videos [17]. However, when the system fails to detect semantic event from the videos due to detection error or noise in a video, those videos will not be considered as a candidate. Our goal is, to retrieve a video with unknown grouped events regardless of systematic errors or missing evidences.

In recent works, Markov Logic Networks (MLN) [14] and Stochastic Context Sensitive Grammar (SCSG) [22] are used for video data representation. SCSG constructs a scene parse graph parsing stochastic attribute grammars. Embodying SCSG, the And-Or graph (AOG) [20] is introduced for scene understanding and can flexibly express more complex and topological structures of the scene, objects, and activities. In this study, we model and represent objects and activities, and their spatial, temporal, and ontological relationships in a scene with And-Or Graph.

When the activities are represented as a graph, finding a similar activity becomes matching similar graphs in video database. Graph matching includes two categories, exact matching and inexact matching. Exact matching requires isomorphism that vertices and connected edges need to be exactly mapped between two graphs or subgraphs. In addition, exact graph matching is NP-complete. On the other hand, inexact graph matching finds mapping between

subsets of vertices with relaxed edge connectivity. It finds suboptimal solution, instead, in polynomial time [7]. The condition for exact matching is quite rigid and makes it difficult to match graphs of videos, where the video, the 2-D projection of 3-D world, has innate noises caused by occlusion, view-point and scale changes, which video analysis methods cannot perfectly handle. In addition, the most of query requires limited retrieval time. Therefore, we apply inexact graph matching. A *substructure similarity search* approach [21] include subgraph matching and indexing algorithm for fast and effective video event retrieval. A complex graph is decomposed into multiple subgraphs. Among the subgraphs, its importance (or selectivity) is determined by frequency over videos. However, the estimation of frequency is rather simple and the same weight is assigned for each sub-graph, which is not distinctive with each other. Therefore we propose to apply other probabilistic methods (e.g. tf-idf [16], pLSA[11], or LDA[2]) to determine the weight of each subgraph from graph database and group the related subgraphs.

LDA is a generative model using Dirichlet prior. LDA has been once widely used for modeling documents [2], scene categorization [9], object recognition [10], and activity recognition [12][19]. For activity recognition, the video is represented by visual features (Spatio-temporal HOG or SIFT) and a complex event is learned from those set of features, called topics (or themes). However recognized activities are mostly simple gestures by a single human (e.g. *running, jumping,* or *boxing*), rather than complex grouped events which involves multiple agents and objects. The main drawback of LDA is that since all features are considered as separate features, the relationships of features are ignored. We plan to apply this topic learning approach while still keeping the relationship of feature pairs.

We propose a novel algorithm to retrieve the semantically closest video from a video query which contains unknown grouped events. The video is analyzed and represented by scene grammars with And-Or Graphs (AOG) [20]. The graph provides a principled mechanism to list visual elements, objects, and activities in the scene and describe their relationships (see Figure 1). These relationships can be spatial, temporal, causal, logical, or ontological. For efficient graph matching, the graph is further decomposed to sub-graphs and then indexed [4]. The sub-graphs are further learned and categorized in unsupervised manner using Latent Dirichlet Allocation (LDA) [2]. The novelty of this study is that: (1) unknown grouped video events with missing evidences are represented by a set of subgraphs; (2) contrasting other subgraph matching algorithms, subgraphs are grouped and matched by indexes after *dimensionality reduction*; and (3) the weights of subgraphs are learned based on their importance in video event corpus. The benefits of our method are: (1) Unknown and untagged grouped events can be matched; (2) Videos with both long and short duration events can be analyzed and matched by semantic reasoning (3) Even though video analyzer fails in finding the correct event, the sub modular activities of the event can be matched to find a similar event; (4) Combination of LDA and subgraph matching reduces disadvantage of each method and boost synergy of their advantages.

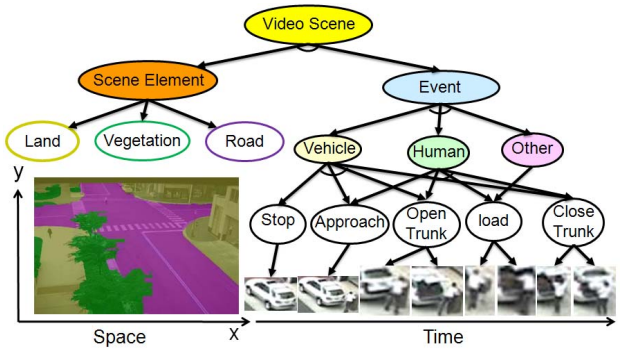


Figure 1. Representation of *loading* event using Spaio-Temporal And-Or graph. The graphical data is indexed by sub-graphs for efficient and robust search.

2. Video Activity Analysis

The And-Or Graph serves as a framework for analysis, extraction, and representation of the visual elements and structure of the scene, such as the ground plane, sky, buildings, moving vehicles, humans, and interactions between those entities. Image content extraction is now formulated as a graph parsing process to find a specific configuration produced by the grammar that best describes the image. The inference algorithm finds the best configuration by integrating bottom-up detection and top-down hypotheses. As illustrated in Figure 1, using a traffic scene as an example, bottom-up detection includes classification of image patches (such as road, land, and vegetation), detection of moving objects, and representation of events, which generate data-driven candidates for scene content. Top-down hypotheses, on the other hand, are driven by scene models and contextual relations represented by the AOG attribute grammar, such as the traffic scene model and human-vehicle interaction model. The fusion of both the bottom-up and top-down approaches results in a more robust video content extraction.

2.1. Scene Element Extraction

Analysis of urban scenes benefits greatly from knowledge of the locations of buildings, roads, sidewalks, vegetation, and land areas. Maritime scenes similarly benefit from knowledge of the locations of water regions, berthing areas, and sky/cloud regions. From video feeds, a background image is periodically learned and it is processed to extract

scene elements. We first perform over-segmentation to divide the image into super-pixels using the mean-shift color segmentation method. Since adjacent pixels are highly correlated, analyzing scene elements at the super-pixel level reduces the computational complexity. For each super-pixel, a set of local features is extracted and super-pixels are grouped by Markov Random Field and Swanson Cut [1]. The example image of extracted scene elements is shown in bottom left of Figure 1. The extracted background scene element helps classification and tracking of a target in the scene after transferred to an action recognition routine.

2.2. Action recognition

The video from the calibrated sensor is processed and metadata of target information is generated by detection, tracking, and classification of targets [4][6]. The metadata consists of a set of *primitives*, each representing target ID, target's classification type, timestamp, bounding box and other associated data for a single detection in a video frame. From metadata, basic actions such as *appear, move*, or *stop* actions are further recognized by analyzing the spatio-temporal trajectory of a target. This is the most time consuming process in the system. To process vast amount of video data, MapReduce framework (http://hadoop.apache. org) is applied to detect basic actions in video data in a distributed system.

2.3. Event Recognition

After recognizing basic actions, event related context is extracted, including: (i) agent (*human, vehicle,* or *general agent*), (ii) basic actions of agent (*appear, disappear, move, stationary, stop, start-to-move*, *turn, accelerate*, *decelerate, etc.*), (iii) properties of events such as *time* (in UTC) and *location* (in latitude/longitude), and (iv) subjects (*human, vehicle, bag, box, door, etc*).
Objects, activities, and spatial (*far, near, beside*) and temporal (*before, after, during, etc.*) relationships are represented by a *parsed graph* after parsing And-Or graphs of complex events. From training data, parameters are learned (for example, threshold values of location and time are learned to determine spatial and temporal relationships), and the structures of And-Or-graphs of the following activities from basic actions to events are built especially for video surveillance applications:

- **Basic action:** *stop/start-to-move, turn, accelerate/ decelerate, hold-bag, carry-box, etc*.
- **Action**: *approach / move-away, lead / follow, catch-up, over-take, meet, etc.*
- **Event**
  - **human-object interaction:**
    - *load / unload*
    - *hand-over*
    - *open/close door/trunk*
  - **human-vehicle interaction:**
    - *embark* / *disembark*
    - *park* (a person disembarks a vehicle and the vehicle remains stationary.) / *ride* (a vehicle was stationary, a person embarks the vehicle, and the vehicle leaves.)
    - *drop-passenger* (a person disembarks a vehicle and the vehicle leaves.) / *pickup-passenger* (a vehicle arrives, a person embarks, and the vehicle leaves.)
    - *loiter-around*
  - **multi-human-vehicle interaction:** *switch-driver, convoy, queuing*.
- **Grouped Events:** combination of multiple events.

Every And-Or graph of listed activities is parsed to infer the events of each video data. We use the simplified *Earley-Stolcke parsing* algorithm [8] to infer an event based on a particular event grammar iteratively. Figure 2 illustrate a parsed graph of a *pick-up* event. When a vehicle appears in the scene and stops, a human approaches the vehicle and disappears, and then the vehicle leaves the scene, this event is defined by the *pick-up* event AOG and represented to the parsed graph. This semantic reasoning may assist videos with both long- and short-term activities to be matched robustly.

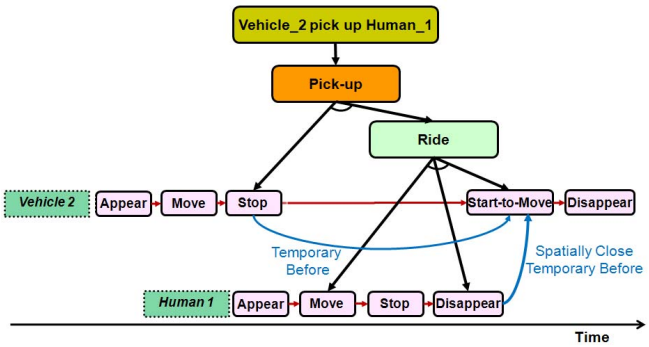


Figure 2. Inference of a *pick-up* event using AOG.

2.4. Group Events

After inferring pre-defined events, a pair of events is again connected by checking spatial or temporal relationship of those events. By doing so, spatially close events or temporally sequential events are connected each other to build a grouped event. This is the important step to keep any unknown event and discover presumably higher-order complex event. The retrieval of those grouped events from the large scale video database is the objective of this work.

3. Video Event Modeling using Sub-Graphs

Inferred scene structure and complex events in videos are represented by relational graphs and saved in database.

When a video is provided as a query, the video is processed to a set of parsed graphs describing activities. Therefore, querying a video becomes querying graphs using similarity search in graph database. This involves matching nodes and edges in graphs with similar attributes and topological structure. As discussed before, exact graph matching requires NP-complete complexity of computing time and yet still be vulnerable to system noises. Therefore, we approach this problem to using inexact graph matching method with subgraph indexing. In spectral graph theory, spectral decomposition is used to represent graphs in a vector space which encodes important structural properties of the graphs. Shearer et al. used subgraph indexing for video retrieval as well [15]. However, similar videos are retrieved by simply finding the largest common subgraph. We propose a new graph indexing method for video-content retrieval. A visual scene can be characterized by a set of subgraph structures. The subgraphs are indexed and further grouped by topics using LDA. The topics are then represented by a feature vector where each entry corresponds to topic distributions of video events. Matching is simply done by vector comparison. As feature vectors are pre-computed for all candidate graphs, searching is very efficient. A schematic illustration of this framework is shown in Figure 3.

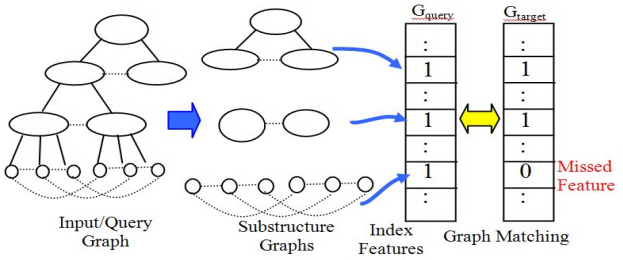


Figure 3. Scalable content indexing and retrieval using graph substructure similarity search. Indexed features are used to represent semantic labels and sub-graphs.

3.1. Subgraph Indexing

All events that occurred in a video are parsed as a graph. A graph G=(**V**, **E**) is defined by a set of nodes **V** and a set of edges **E**. In the graph, each node $v \in$ **V** represents an agent or an event, and each edge $e \in$ **E** corresponds to the relationship between (1) two objects (*e.g. ontological relationship*), (2) an event and an object (e.g. *has relationship*), or (3) two events (e.g. *spatio-temporal relationships*). An example of a parsed graph is shown in Figure 4 illustrating "*a vehicle stops and a person comes out*."

Then, the graph G is decomposed to subgraphs. First the node is selected for one-node subgraphs, then two-nodes connected with an edge are extracted for two-node subgraphs, and then n nodes connected by edges are formed for n-node subgraphs. Figure 5 shows the decomposed subgraphs of a single graph in Figure 4. Figure 5-(a) shows one-node subgraph, Figure 5-(b) shows two-node subgraphs, and Figure 5-(c) shows three-node subgraphs. After that, each subgraph is indexed and saved in a subgraph feature vocabulary.

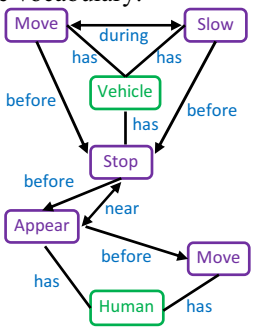


Figure 4. A parse graph for a complex event (d*isembark* event)

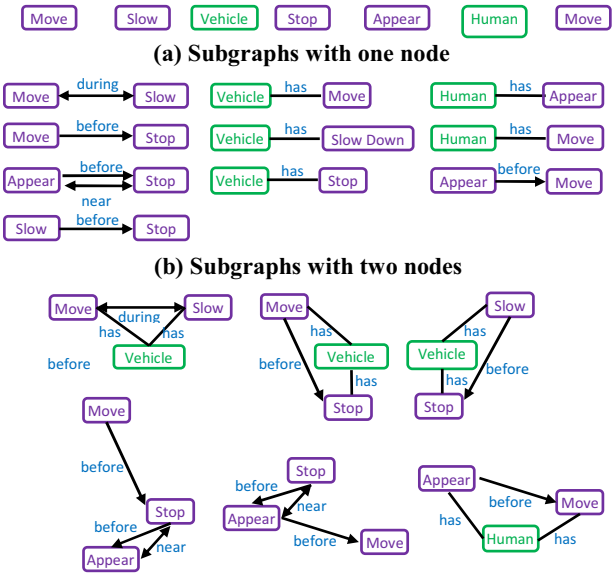


Figure 5. Example of a set of subgraph features from the graph in Figure 4.

3.2. Subgraph Matching

After converting video data to a graph, the video event search problem between query video and videos in database becomes the graph matching problem. Given a query graph, G_q, finding the closest graph from graphs in database, DB, is determined by maximizing energy function E.

$$Q(G_q) = \max_{r \in DB} E(G_q, G_r) \qquad (1)$$

where G_r is one of graphs in the video repository DB. A graph with maximum energy is selected as a matching graph. Now we define the energy function E as subgraph matching:

$$E(G_q, G_r) = \sum_{a \in g_q, g_r} \theta_a x_a + \sum_{a,b \in g_q, g_r} \theta_{ab} x_{ab} + \ldots + \sum_{a,b,c \ldots n \in g_q, g_r} \theta_{abc \ldots n} x_{ab \ldots n} \quad (2)$$

where E is the correspondence energy between two graphs, G_q and G_r. g_q is a set of subgraphs of G_q and g_r are subgraphs of G_r. $x \in (0,1)$ ($x=1$ when there is matching subgraph in both G_q and G_r, $x=0$ otherwise) indicates corresponding subgraph features with one node x_a, two nodes x_{ab} and n nodes $x_{ab \ldots n}$ in both G_q and G_r. θ is a weight for the correspondence.

In Equation (2), the graph matching problem is decomposed by matching subgraphs with one node (first term), two nodes (second term) or n nodes (last term). More nodes in subgraph represent more complex relationships among the nodes. However, computational time and the number of subgraphs increase exponentially as the node size increases. More subgraphs can have more redundant and conceptually duplicated subgraphs. In experiment results in Figure 11, subgraphs with one and two nodes were optimal on performance, speed, and memory for video event search.

After indexing subgraphs, the equation becomes much simpler since a set of subgraphs in a video are represented by a vector.

$$E(G_q, G_r) \approx E(g_q, g_r) = \sum_{\substack{q_i \in g_q, r_i \in g_r, \\ s=1..S}} \theta_s x(q_s, r_s) \quad (3)$$

where q_s is an indexed subgraph in a query video, r_s is an indexed subgraph in database, the size of subgraph vocabulary is S, $x(q_s, r_s)=1$ when both q_s and r_s exist, 0 otherwise.

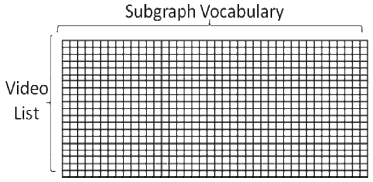
In Equation (3), the most important factor is θ. When a node is a visual feature, θ can be appearance measure (shape context, SIFT, HOG, or color histogram, or bag-of-words in a bounding box of human, vehicle, or object) or geometric distance. When a node is a semantic node, θ can be ontological distance (the distance in an ontological family tree such as WordNet) or importance of the subgraph itself.

Since a node represents semantic context in our case, we use θ as weight or importance of subgraphs. Rather than having one θ value for a corresponding subgraph, we set different values with respect to each video. We learn such θ from the corpus of video database, applying tf-idf, pLSA, and LDA.

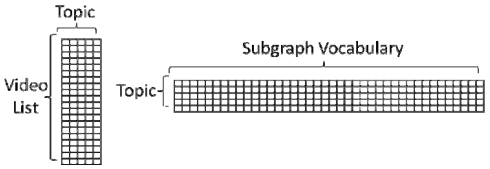
Tf-idf finds relationship between word and document using frequency in a document and inverse document frequency in a discriminative manner. In our application, tf-idf builds a subgraph-by-video matrix which defines correlation θ between subgraphs and videos.

$$\theta_{sv} = \frac{f_{sv}}{\max_{w \in v} f_{wv}} \cdot \log \frac{V}{\left| \{ v \in \mathbf{V} : s \in v \} \right|} \quad (4)$$

Where $\mathbf{V}$ is video corpus and V is its number. f_{sv} is frequency of subgraph s in video v. The first term is subgraph frequency and the second term is inverse video frequency. Unlikely having constant θ over a video as shown in Equation (3), frequency and video related matrix θ is defined. The matrix is shown in Figure 6-(a). However, the constructed matrix is too large and characteristic of documents are not captured.



(a) A subgraph-by-video matrix of weight parameter θ in tf-idf. The dictionary size is SV.



(b) A subgraph-by-topic matrix and topic-by-video matrix in pLSA or LDA. The dictionary size is (ST+VT).

Figure 6. Illustration of database size comparison between tf-idf and LDA or pLSA.

To reduce the large scale matrix and find characteristics of each video, pLSA is introduced [11]. In pLSA, a video is modeled by a set of latent variables (so called topics) which is built from Gaussian mixture of subgraphs. This mixture model divides a big subgraph-by-video matrix to two smaller matrices, subgraph-by-topic and topic-by-video. However, pLSA has an issue such that the number of parameters increases as data size increases, which may cause overfitting and requires more time for re-learning new dataset [2].

To overcome pLSA's issues, LDA is designed. Like pLSA, LDA also reduces dimension, and model the topics as shown in Figure 6-(b). Besides, generative semantic meanings are modeled from a set of video and subgraphs. Another main advantage of LDA is that when a new video is added in database, update of the system is much faster and simpler than other methods. Applying LDA, the energy function is further simplified to compare topics rather than all subgraphs. In LDA, topic distribution $\boldsymbol{\theta}_v = \{\theta_{v1}, \theta_{v2}, \ldots, \theta_{vt} \ldots, \theta_{vT}\}$ is learned, where θ_{vt} represents relationship between video and topics. The learned dictionary is illustrated in Figure 6-(b)-left. The other parameter, θ_{ts}, represents relationship between topics and subgraphs (See Figure 6-(b), right). We infer these parameters using EM-algorithm as discussed in [2].

Using LDA, all subgraphs are transferred to topics and topics are, again, indexed and modeled in a topic vector. After all, subgraph matching is simply done by comparing topic distribution over videos.

$$E(G_q, G_r) \approx E(T_q, T_r) = Dist(\boldsymbol{\theta}_q, \boldsymbol{\theta}_r) \quad (5)$$

where $\boldsymbol{\theta}_q$ is topic distribution vector of G_q and $\boldsymbol{\theta}_r$ is topic distribution vector of G_r. $Dist(\cdot)$ is the distance function between $\boldsymbol{\theta}_q$ and $\boldsymbol{\theta}_r$. The distance function can be L-1, L-2, Chi square or earth mover's distance. The performance of each distance function is discussed in next section.

4. Experimental Results

We implemented a video event search system that accepts video as a query and provides closest videos in a sorted order. For learning, a video is fed to multiple processors and processed to detect scene elements, actions, and complex events by parsing AOG. The detected events are described in parsed graphs and subgraphs are built and indexed. Those processes are performed in a distributed computing system for the fast and reliable system. After that, the subgraph features are learned to extract topic and learn parameters. The learned parameters are used for video-to-video search. The pipeline is shown in Figure 7.

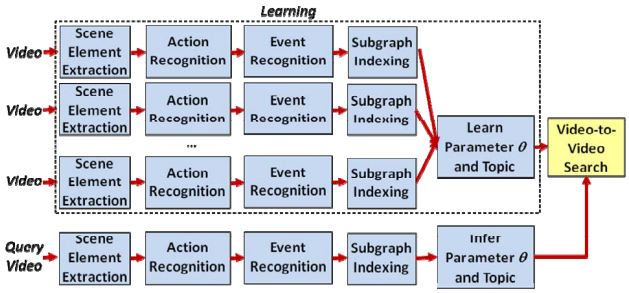


Figure 7. Pipeline of video event search engine.

In our knowledge, there was no baseline method for complex grouped video events retrieval or those dataset for evaluation, even though the need of video event search increases. Using one node subgraph can be considered as bag-of-word based method as a baseline method. TRECVID dataset (http://trecvid.nist.gov) is widely used for video retrieval evaluation. However, the dataset focuses on detection of basic actions (e.g. *PersonRun*, *CellToEar*, or *Pointing*) or scene categorization rather than recognition of grouped events. For that reason, TRECVID dataset does not fit in this evaluation. Therefore, we used 262 web-collected surveillance video clips including VIRAT dataset containing grouped events [13]. The play time of each video clip is around 2 minutes and they are mostly taken at different sites in different times. Among them, 212 videos are selected for training and database videos and other 50 video clips, from which majority of human annotators could select their closest video in database, are selected as test query videos. In the query videos, the events includes from basic actions (e.g. "*vehicle-passing-by*") to grouped events (*e.g.* "*vehicle park, a human_A get off, unload box, human_B meet human_A, human_A hand over a box to human_B, human_B disappear, human_A ride the car, the car disappear.*"). We plan to have this dataset available to the public.

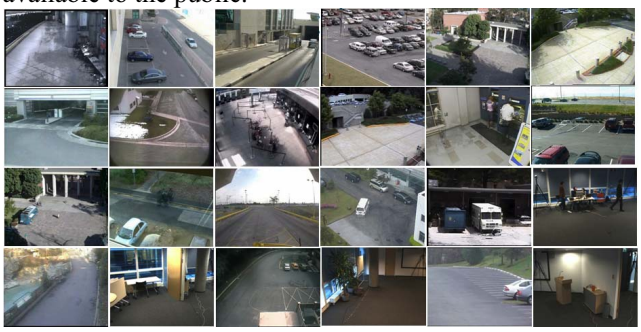


Figure 8. Snapshots of some test videos

After processing training video dataset, the number of one node subgraph was 33, that of two node subgraphs was 1384, and that of three node subgraph was 37431 as shown in Figure 9.

Single node (33)

HUMAN	SEDAN
PICKUP_TRUCK	
LAND_VEHICLE	
drops_passenger	
passes_object	
meets	moves
disembark	follows
convoys	unloads
...	

Two nodes (1384)

meets+hasPatient+HUMAN
follows+hasPatient+HUMAN
moves+During+meets
turns+Before+meets
follows+During+meets
loads+spatialNear+rides
embarks+hasPatient+LAND_VEHICLE
rides+hasPatient+LAND_VEHICLE
...

Three nodes (37431)

leads+Before+moves:leads+hasPatient+SEDAN
follows+hasPatient+HUMAN:follows+Before+stops
decelerates+hasPatient+HUMAN:decelerates+Starts+follows
leads+Equals+follows:leads+Before+follows
convoys+Starts+leads:convoys+Equals+convoys
...

Figure 9. The example of subgraphs with one, two, and three nodes from video surveillance data.

Topic 11	Topic 14	Topic 76
HUMAN	follows+hasPatient+LAND_VEHICLE	moves+hasPatient+LAND_VEHICLE
OTHERS	leads+hasPatient+LAND_VEHICLE	moves+Meets+moves
LAND_VEHICLE	Leads+During+stays_stationary	turns+hasPatient+LAND_VEHICLE
moves+hasPatient+LAND_VEHICLE	follows+During+stays_stationary	turns+Finishes+moves
moves+Before+parks	follows+During+moves	turns+Before+moves
disembarks+During+move	moves+During+moves	moves+Meets+turns
disembarks+During+stationary	leads	turns
moves+Before+stays_stationary	leads+During+moves	turns+Starts+moves
moves+Before+stops	moves+During+stays_stationary	moves

Figure 10. Example of topics and nine most relevant subgraphs for each topic.

We evaluated performance with different topic size from 10 to 1000, the performances were quite similar but 100 topics gave the best result. Therefore, we set topic size to 100. The example of extracted topics after applying LDA is shown in Figure 10. *Topic_11* consists of subgraphs with events with human and vehicle. *Topic_14* is about vehicles' lead/follow events, *Topic_76* consists of a vehicle's turning events.

We also evaluated our video event retrieval algorithm using subgraph indexing with different (1) subgraph node sizes (2) weighting and grouping schemes with tf-idf, pLSA, and

LDA, and (3) distance functions. We conducted experiments with all three dimensions, but some of them are shown here for clearer display.

Experiment 1: Different subgraph node sizes

The retrieval rate with different node size is shown in Figure 11. The retrieval rate shows the correct matching rate between query video and corresponding groundtruth video as the retrieved rank increases. Using one-node subgraphs as features can be considered as a general bag-of-word based approach. Using two-node subgraphs denotes retaining relationship between two nodes.

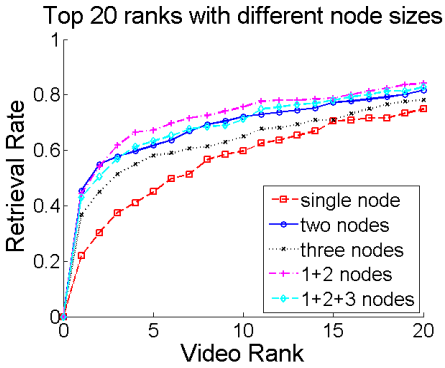


Figure 11. Retrieval rate using 5 different combinations of subgraphs' node size (one, two, three, one+two, and one+two+three nodes) using LDA.

From the evaluation results, we could observe that the bag-of-word method with a single node gives worst results as the relationship of nodes is ignored. On the other hand, the subgraphs with a single node and two nodes gave best results. The performance gets slowly worse as node's size increases. The larger size of nodes captures higher-order relationships but exponentially increased subgraphs are more conceptually duplicated each other and become less discriminative across video corpus. An application, which requires more complex relationships among nodes, may require subgraphs with more nodes. We conducted the experiments with tf-idf and pLSA with varying node sizes and they provided the same trend, where one+two nodes gave the best retrieval rate.

Experiment 2: tf-idf, pLSA, and LDA

The performance of tf-idf, pLSA, and LDA are shown in Figure 12. The experimental results show that LDA models video events best among three of them. Using LDA with 1+2 nodes, 22 out of 50 (44%) videos are correctly retrieved as a first rank and 40 videos (80%) are correctly retrieved within top 20 ranks, which can be shown in a first page in our browser-based video retrieval system. Other 10 videos retrieved with lower ranks were videos containing only common events which most of database videos contain such as *car-passing-by* or *human-walk*.

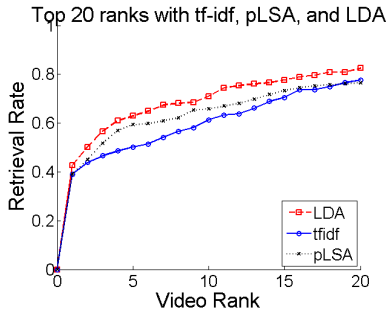


Figure 12: Retrieval rate of LDA, tf-idf and pLSA with 1+2 nodes are compared as video rank is increased.

Experiment 3: Distance functions

We compared five different distance functions of LDA's topic distributions or tf-idf's subgraphs in Equation (5), Euclidean, Earth mover distance, Cosine, L1 and Chi square. Their performances are shown in Figure 13. LDA with Chi square and L1 distances gave the best results among 5 distance metrics. The results were similar for pLSA.

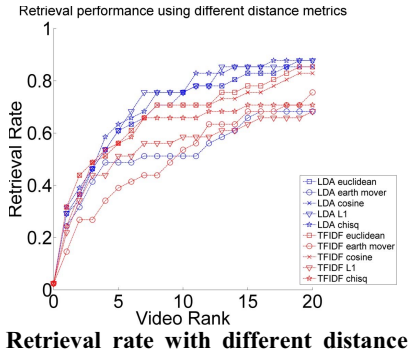
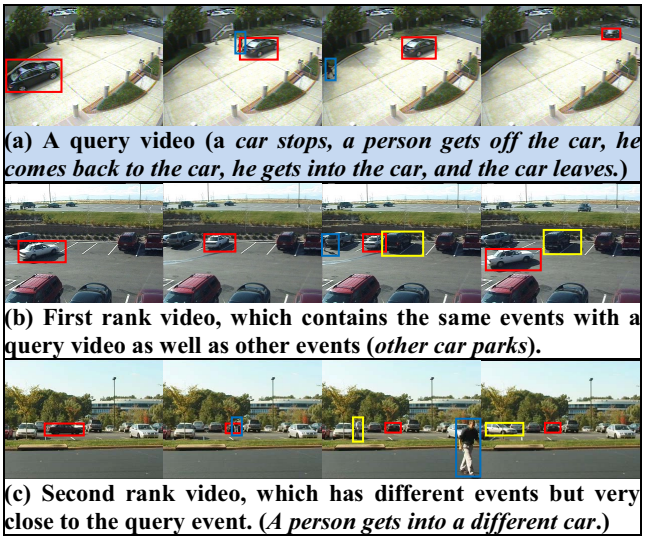


Figure 13. Retrieval rate with different distance functions, Euclidean, Earth mover distance, Cosine, L1 and Chi square with either LDA or tf-idf.

Examples of query and best two matching videos are shown in Figure 14. The query video contains, "*a car appears, the car stops, a human dismounts the car, the human comes back to the car, the human mounts the car, the car goes away*." as shown in Figure 14-(a). After subgraph matching, the first rank video is shown in Figure 14-(b), which has exactly the same event including some other events ("*other vehicles are parked.*"). The second rank video is in Figure 14-(c), which has quite similar events however two vehicles and two persons are involved. Since substructures of a graph are matched, a set of graphs with similar subgraphs can be extracted with high matching score. However, since Figure 14-(b) keeps more structurally similar relationships among the nodes, it gets higher score than a set of subgraphs in Figure 14-(c).

The average time of processing a query video was around 10 minute for 2 minute video using 2.8 GHz Intel Xeon CPU spending most of times on video analysis and basic action recognitions. For the pre-processed query videos, the retrieval time was less than 1 second over networks.



(a) A query video (a *car stops, a person gets off the car, he comes back to the car, he gets into the car, and the car leaves.*)

(b) First rank video, which contains the same events with a query video as well as other events (*other car parks*).

(c) Second rank video, which has different events but very close to the query event. (*A person gets into a different car.*)

Figure 14: Snapshots of query and retrieved video clips. The yellow bounding boxes indicate noisy data.

5. Conclusion

A novel framework for video modeling and video event search has been developed. The closest grouped events are retrieved even with noises from data and missing detections. Video event retrieval is converted to a graph matching problem and solved using novel subgraph grouping and matching. Topics of each subgraph are modeled by a generative probabilistic framework and relevance and importance of topics are learned over videos. The experimental results show that subgraphs with one and two node sizes, LDA for learning, and Chi square distance function over topic distributions provided best performance. The method and system were robust with most of both short and long duration videos. When the event of interest is spatially and temporally scattered (*e.g. a human take a bag in one city and put it down in the other city 2 hours later.*), those spatial and temporal relationships needs to be connected for recognition. The experimental data was mostly surveillance videos. Nonetheless, the method can be applied to any type of videos such as news, movies, or personnel video clips when their video analysis methods are available. The proposed method can be also extended to matching on any graphical structure. Future research includes combining user's text description of a video and automatically extracted semantic context for video event search. Other research contemplates extending the subgraph indexing algorithm to other applications such as multiple target tracking across multiple cameras.

Acknowledgement

This material is based upon work supported in part by the Office of Naval Research under contract number N00014-10-C-0308 and DARPA under contract number FA8650-11-1-7149.

References

[1] A. Barbu, S.C. Zhu, "Graph partition by Swendsen-Wang cut," *ICCV*, 2003.

[2] D. Blei, A. Ng, M. Jordan, "Latent Dirichlet allocation". Journal of Machine Learning Research, 3:993-1022, 2003.

[3] C.F. Chang, W. Chen, H.J. Meng, H.Sundaram, D. Zhong, "A Fully Automated Content Based Video Search Engine Supporting Spatio-Temporal Queries," PAMI, 1998.

[4] T.E. Choe, M.W. Lee, N. Haering, "Traffic Analysis with Low Frame Rate Camera Network", WCN2010.

[5] T.E. Choe, M.W. Lee, F.Guo, G. Taylor, L. Yu, N. Haering, "Semantic Video Event Search for Surveillance Video", Workshop on Visual Surveillance (VS2011), 2011

[6] T.E. Choe, et al, "Globally Optimal Target Tracking in Real Time using Max-Flow Network", VS2011.

[7] D. Conte, P. Foggia, C. Sansone, M. Vento, "Thirty Years Of Graph Matching In Pattern Recognition," Int. Journal of Pat. Rec. and Art. Int., Vol. 18, No. 3, pp. 265-298, 2004.

[8] J. Earley, "An efficient context-free parsing algorithm", *Communications of the Association for Computing Machinery*, 13:2:94-102, 1970.

[9] L. Fei-Fei, P. Perona, "A Bayesian Hierarchical Model for Learning Natural Scene Categories," CVPR 2005.

[10] R. Fergus, L. Fei-Fei, P. Perona, A. Zisserman, "Learning object categories from google's image search," IEEE International Conference on Computer Vision, 2005.

[11] T. Hofmann, "Probabilistic latent semantic indexing," Proceedings of the Twenty-Second Annual International SIGIR Conference, 1999.

[12] J.C. Niebles, H.Wang, L. Fei-Fei, "Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words," IJCV 2008.

[13] S. Oh et al., "A Large-scale Benchmark Dataset for Event Recognition in Surveillance Video," CVPR 2011

[14] M. Richardson, P. Domingos "Markov logic networks." *Mach. Learn.*, 62:107–136, 2006.

[15] K. Shearer, H. Bunke, S. Venkatesh, "Video indexing and similarity retrieval by largest common subgraph detection using decision trees," Pattern Recognition, 2001

[16] Slaton, McGill, editors, "Introduction to Modern Information Retrieval," McGraw-Hill, 1983.

[17] C.G.M. Snoek, B Huurnink, L Hollink, M.D. Rijke, G. Schreiber, M. Worring, "Adding semantics to detectors for video retrieval," *IEEE Trans. on Multimedia*, 2007.

[18] C.G.M. Snoek, M. Worring, "Multimedia Event-Based Video Indexing Using Time Intervals," *IEEE Trans. on Multimedia*, Vol.7, NO.4, AUGUST 2005.

[19] Y. Wang , P. Sabzmeydani , G. Mori, "Semi-latent Dirichlet allocation: A hierarchical model for human action recognition", Workshop on Human Motion Understanding, Modeling, Capture and Animation, 2007.

[20] T.Wu,S.Zhu,"A Numeric Study of the Bottom-up Top- down Inference Processes in And-Or Graphs," *ICCV*, 2009.

[21] X. Yan, P.S. Yu, and J. Han, "Substructure Similarity Search in Graph Databases," *SIGMOD*, June 2005.

[22] S.C. Zhu, D. Mumford, "Quest for a stochastic grammar of images", *Foundations and Trends of Computer Graphics and Vision*, vol.2, no.4, pp259-362, 2006.