

Relational Graph Mining for Learning Events from Video

Muralikrishna Sridhar and Anthony G Cohn and David C Hogg¹

Abstract. In this work, we represent complex video activities as one large activity graph and propose a constraint based graph mining technique to discover a partonomy of classes of subgraphs corresponding to event classes. Events are defined as subgraphs of the activity graph that represent what we regard as *interesting* interactions, that is, where all objects are actively engaged and are characterized by frequent occurrences in the activity graph. Subgraphs with these two properties are mined using a level-wise algorithm, and then partitioned into equivalence classes which we regard as event classes. Moreover, a taxonomy of these event classes naturally emerges from the level-wise mining procedure. Experimental results in an aircraft turnaround apron scenario show that the proposed technique has considerable potential for characterizing and mining events from video.

1 Introduction

An important problem in computer vision is to learn a high level understanding of *complex activities* from videos starting with low level visual analysis. Such an understanding involves learning the *events* which are the natural building blocks of activities, and also their structural partonomic relationships. Complex activities are usually composed of multiple events that may occur in parallel, and overlapping events may share participating objects. Complex activities also contain spurious and missing objects and spatial relationships, arising either due to instability in image processing or due to coincidental occurrences. We address the problem of *unsupervised discovery* of an event partonomy from such complex video scenes.

An important problem in graph mining is to mine interesting subgraphs from a graph database or a single graph. Several techniques [2] have been developed to mine subgraphs that are interesting either because of their frequency or for satisfying certain constraints. In this work, we represent activities as a single large *activity graph*. The key hypothesis is that events (in contrast to noise and coincidental occurrences) correspond to interesting subgraphs of this activity graph and are hence called *event graphs*.

Our earlier work [9] introduced a relational qualitative spatio-temporal representation called an *activity graph* to represent interactions between all objects in a scene. Two measures of interestingness - *frequency* and a *manually defined focus mechanism* were used to drive the mining process for discovering event graphs. We have very recently improved the representation in [11] with a more robust variable free activity graph and a generic focus mechanism called *interactivity*, both of which we adopt in this work. In [11], we focussed on learning the most probable interpretation of a video using a generative model. In this work, we adopt a complementary graph mining approach of learning an event partonomy by characterizing *events* as

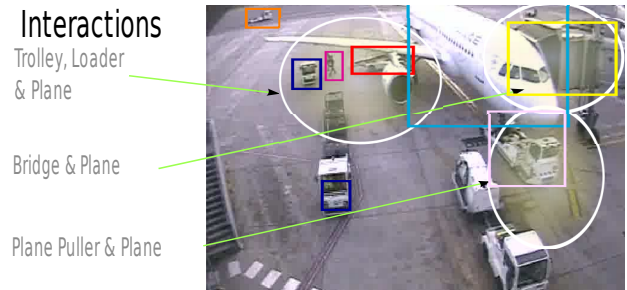


Figure 1. Aircraft handling scenario. The highlighted ellipses shows some groups of interacting objects.

sufficiently *frequent* and *interactive* subgraphs of the activity graph. The underlying hypothesis is that non-events which may be observation noise or coincidences do not tend to possess these systematic properties. This hypothesis is validated on large video data set capturing activities in an aircraft apron.

This paper presents a more formal treatment of the graph mining technique that has been very briefly introduced in our recent short paper in [10], where also a HMM for robustly computing the activity graph is introduced. This paper also formalizes the interactivity measure in terms of graphs, which was originally formulated in terms of tracks [11].

2 Related Work

Much previous work on event analysis represents activities as propositional sequences rather than in a more expressive relational form such as logic or graphs. Sequential representation has been used for unsupervised learning of events using standard frameworks such as pattern recognition techniques [14], graphical models [13] and grammars [7]. However, activities that are composed of events happening in parallel or with shared objects are challenging to mine with sequence based representations [3] or even those that may use logical sequences [1], since sequences do not form a natural representation for such parallel overlapping activities.

These problems are addressed in [9], where we introduced a relational graph based representations of activities for representing interactions between objects. This representation has been further modified in [11] with a more robust variable free activity graph where graph mining frameworks can be directly applied. We also introduced a generic focus mechanism called *interactivity*, both of which we adopt in this work. The following paragraphs provide an overview of graph mining approaches related to this work.

Much of the initial work on graph based learning [15] focussed on frequent subgraphs since the isomorphism of graphs is combinatorially expensive [6]. Despite this restriction, many solutions that efficiently search the space of candidate frequent graphs have been

¹ University of Leeds, UK, {krishna,agc,dch}@comp.leeds.ac.uk. This work is supported by the EPSRC (EP/D061334/1) and the EU FP7 (Project 214975, Co-Friend). We also thank colleagues in the Co-friend project.

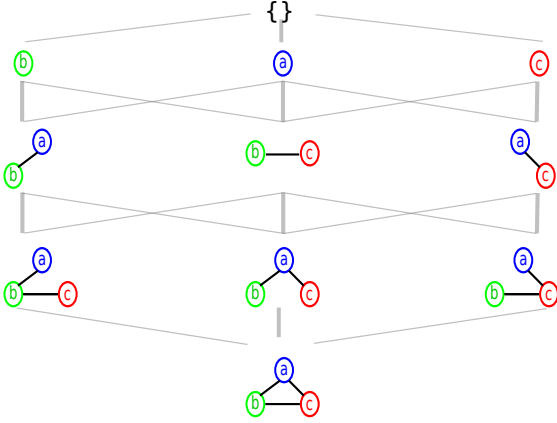


Figure 2. This edge expansion lattice is the search space for mining graphs. In this lattice, each level k corresponds to graphs with k edges.

developed. The search space of frequent candidate graphs can be organized as an edge expansion lattice shown in Fig. 2 and graph mining techniques tend to exploit the structure of this lattice in order to efficiently steer the search for frequent or more generally interesting subgraphs.

The authors in [4] introduced SUBDUE, which is a greedy beam search based technique that uses the MDL principle for obtaining a compressed representation of an input graph(s). However, this approach is prone to getting stuck in a local optimum as Subdue is based on a greedy search with no backtracking. Depth first approaches such as Gspan[17] dramatically improves the performance by reorganizing the edge expansion lattice into a DFS code tree, where the nodes at the level k corresponds to a candidate subgraph with k edges.

In this work, we adopt a breadth first approach. These approaches, such as AGM [5], FSG [8], tend to search the space to generate candidate $k + 1$ sized graphs by combining *only* pairs of frequent k sized graphs, that share a common $k - 1$ size graph. The frequency of a $k + 1$ size candidate graph is computed by scanning the graph data base or a single graph. By using only frequent graphs to generate candidates at the next level, the search space is kept under control.

The techniques above have been developed mainly for improving scalability on subgraph mining. However, when the graphs are dense, the extraction is not always tractable and results in many uninteresting graphs being generated. CabGin [12] and gprune [19] were paradigms that were introduced to mine constraints in order to reduce the cost of mining and increase the focus on interesting patterns.

However, from a literature survey [2] and experimentation with existing techniques for constraint based graph mining, we have not found that these existing constraints allow us to naturally mine for the desired structures in our domain of application. In this work we use the breadth first approach for searching a lattice of candidate event graphs using two interestingness measures – frequency and interactivity – to drive the search.

3 Qualitative Spatio-Temporal Graphs

This section describes the representation of a video activity in terms of an activity graph. The activity graph that represents interactions between all the tracks in a scene. The *key feature* of the activity graph is that it abstracts the spatio-temporal relationships between tracks, away from other metric details of their interaction, such as the spatial locations, temporal duration and object features. Thus, the activity graph facilitates comparison of interactions, since spatio-temporally

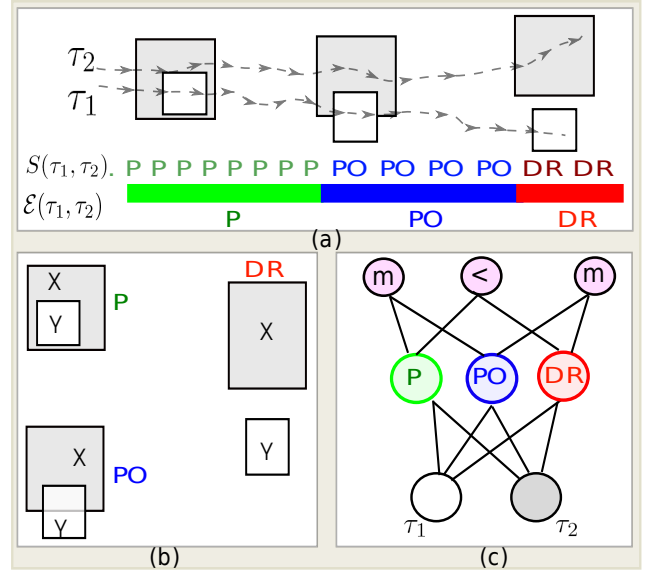


Figure 3. (a) Two tracks, τ_1, τ_2 , sequence of spatial relations $S(\tau_1, \tau_2)$ and episodes $\mathcal{E}(\tau_1, \tau_2)$. (b) Spatial relations $\{P, PO, DR\}$. (c) Activity graph for the interaction between τ_1, τ_2 .

identical (resp. similar) interactions induce isomorphic (resp. similar) subgraphs in the activity graph.

Tracks. Given a video, object tracks $T = \{\dots, \tau_i, \dots\}$ (e.g. τ_1, τ_2 in Fig. 3.a) are obtained using techniques in [18]. For the purpose of expositional simplicity, we use a scene with just two tracks τ_1, τ_2 shown in Fig. 3.a and represent their interaction in an activity graph shown in Fig. 3.c.

Spatial Relations. For each pair of tracks (e.g. τ_1, τ_2), a sequence (e.g. $S(\tau_1, \tau_2)$ in Fig. 3.a) of qualitative spatial relations, which are either $\{P, PO, DR\}$ (Fig. 3.b) are computed using a HMM, as detailed in [10].

Episodes. For each pair of tracks τ_i, τ_j , the sequence of spatial relations is aggregated to a sequence of *episodes* (e.g. $\mathcal{E}(\tau_1, \tau_2)$ in Fig. 3.a), such that within the temporal interval of each episode, the same spatial relation holds, but a different spatial relation holds immediately before and after this interval. The set of all episodes between the tracks for an activity is given by \mathcal{E} .

We define the following functors on episodes (e, e') for later use: (i) $Tracks(e) = \langle \tau_i, \tau_j \rangle$ maps to the respective pair of tracks for e ; (ii) $\mathcal{I}(e)$ maps to the temporal interval for e ; (iii) $Spatial(e) \in \mathfrak{R}$ maps to the spatial relation for e ; (iv) $Temporal(\mathcal{I}(e), \mathcal{I}(e'))$ maps to Allen’s temporal relationship between the intervals corresponding to two episodes e, e' .

Activity Graph. An activity graph $G = (V, E, \rho, \eta, \mathfrak{R}, \mathfrak{S})$ is a directed edge-labelled *layered graph* in which the vertices V are partitioned into 3 layers V^1, V^2, V^3 and edges E exists only between adjacent layers. All our relations are binary, so there are exactly two edges from each node to nodes in the layer below. The function ρ maps the nodes in the second layer to labels which are spatial relations in \mathfrak{R} . The function η maps nodes in the third layer to labels which are Allen’s temporal relations in \mathfrak{S} . The activity graph is described more precisely below.

Layer 1 nodes of the activity graph correspond to tracks (e.g. τ_1, τ_2 in Fig. 3.a). That is, there is a 1-1 mapping $\gamma_1 : V^1 \leftrightarrow \mathcal{T}$. However, these nodes are not explicitly labelled with any details of these tracks, in order to abstract away information specific to these tracks.

Layer 2 nodes of the activity correspond to episodes (e.g. $\mathcal{E}_{\tau_1, \tau_2}$ in

Fig. 3.a). That is, there is a 1-1 mapping $\gamma_2 : V^2 \leftrightarrow \mathcal{E}$ where \mathcal{E} is the set of episodes generated by the tracks $T = \gamma_1(V^1)$. These nodes are labelled with spatial relations between the respective pairs of tracks pointed to at layer 1 as shown in Fig. 3.c. So we define a mapping $\rho : V^2 \rightarrow \mathfrak{R}$ such that for $v \in V^2$, $\rho(v) = s \in \mathfrak{R}$ is equivalent to:

$$s = \text{Spatial}(\gamma_2(v)) \wedge \text{Tracks}(\gamma_2(v)) = (\gamma_1(v'), \gamma_1(v'')) \\ \wedge v' \in V^1 \wedge v'' \in V^1 \wedge \langle v, v' \rangle \in E \wedge \langle v, v'' \rangle \in E$$

Layer 3 nodes of the activity graph relate pairs layer 2 episode nodes. That is, there is a 1-1 mapping $\gamma_3 : V^2 \times V^2 \leftrightarrow V^3$. These nodes are labelled with Allen's temporal relations (e.g. m : meets, $<$: before in Fig. 3.c) between intervals corresponding to the episodes for pairs of layer 2 nodes. So we define a mapping $\eta : E \rightarrow \mathfrak{S}$ such that for any $v \in V^3$, $\eta(v) = t \in \mathfrak{S}$ is equivalent to:

$$t = \text{Temporal}(\mathcal{I}(\gamma_2(v')), \mathcal{I}(\gamma_2(v''))) \wedge \gamma_3(v) = (v', v'') \\ \wedge v' \in V_2 \wedge v'' \in V_2 \wedge \langle v, v' \rangle \in E \wedge \langle v, v'' \rangle \in E$$

Graphs or Logic ? In this work, we have adopted a graph based representation of interactions, since several frameworks for efficiently mining from graphs have been developed [2] in the data mining research community. Equivalently, interactions can also be expressed as logical formulae and logic based relational data mining techniques could be potentially applied. For example, the interaction in Fig. 3.a with X, Y as object variables and I_1, I_2, I_3 as temporal variables can be represented logically as:

$$\text{holds}(P(X, Y), I_1) \wedge \text{holds}(PO(X, Y), I_2) \wedge \text{holds}(DR(X, Y), I_3) \\ \wedge \text{meets}(I_1, I_2) \wedge \text{meets}(I_2, I_3) \wedge \text{before}(I_1, I_3)$$

Note the need for variable repetition in the textual formula compared to the variable free graphical representation where variable co-occurrence is represented implicitly by the unique nodes for each object/episode and multiple parent nodes.

4 Event Definition

The activity graph represents an exhaustive set of spatio-temporal relationships between all the tracks from a video. We are interested only in candidate event graphs which are defined as those subgraphs of the activity graph that correspond to semantically meaningful interactions between a set of tracks. Event graphs are defined as those candidate event graphs which are also *maximally frequent* and *sufficiently interactive*. We apply a breadth first graph mining technique detailed in section 4.2 to discover classes of isomorphic event graphs, which are regarded as event classes. The following paragraphs describe candidate event graphs and event graphs before proceeding to the mining technique.

4.1 Candidate Event Graphs.

Candidate event subgraphs of the activity graph are intended to correspond to the conceptual notion of an interaction between a set of tracks. A subgraph of the activity graph is a candidate event subgraph if there exists some interval \mathcal{I} such that it describes all of the spatio-temporal interactions between a set of tracks during \mathcal{I} (and only those interactions). That is to say that, firstly every episode (layer 2 node) involving any of the tracks and temporally connected to \mathcal{I} is present

in the subgraph.

$$\forall e \in \mathcal{E} : \text{Tracks}(e) \in T \wedge \mathcal{I}(e) \cap \mathcal{I} \neq \emptyset \Rightarrow \\ \exists v \in V^2 : \gamma_2(v) = e$$

Secondly, all temporal relationships involving these episodes are represented as layer 3 nodes in the subgraph, except between those pairs of episodes which are both either (i) initial episodes \mathcal{E}_i for a pair of tracks, or (ii) both final episodes \mathcal{E}_f for a pair of tracks.

$$\forall v', v'' \in V^2 : \langle v', v'' \rangle \notin \mathcal{E}_i \wedge \langle v', v'' \rangle \notin \mathcal{E}_f \Rightarrow \\ \exists v \in V^3 : \gamma_3(v) = \langle v', v'' \rangle$$

This latter condition is necessary to ensure isomorphism of similar events, which only differ in when the initial and final episodes respectively start and end².

Maximally Frequent. A candidate event graph H is *frequent* with respect to an activity graph, iff the number of node-isomorphic instances of H in the activity graph is greater than a threshold λ_1 . A frequent layered graph H is also *maximally frequent* if there is no other proper super graph H' of H , which is as frequent as H . The value of the thresholds λ_1 for maximally frequent and λ_2 for sufficiently interactive criterion as detailed below, are described further in section 4.2.

Sufficiently Interactive. As already identified in the introduction, we are more interested in candidate events in which all participating tracks are actively engaged uniformly over time. More precisely, we prefer those candidate event graphs in which spatial relations are distributed uniformly (i) *across all subsets of tracks* (for e.g. Fig. 4 (a) rather than Fig. 4 (b)) (ii) *temporally* (for e.g. Fig. 4 (a) rather than Fig. 4 (c)). Preference (i) means preferring candidate events with fewer tracks involved ignoring extraneous ones. (ii) means preferring candidate events in which interactions between pairs of tracks are tightly interleaved. The following details may be omitted on a first reading.

Pointwise mutual information (PMI)[16] is a well suited measure to model the degree of association between a subset of outcomes belonging to random variables. We model the degree of interaction between a subset of tracks for a candidate event, in terms of the PMI between them. Then we express *interactivity* in terms of pointwise total correlation [16], which is just a weighted sum of PMIs over all subsets of tracks for a candidate event graph. Pointwise total correlation is highest when interactions between the tracks for the candidate event graph H are well distributed, both temporally and amongst subsets of tracks.

Let U_1, U_2, U_3 be the nodes corresponding to the three layers of the a candidate subgraph H of the activity graph. PMI measures the strength of association between a set of tracks ε corresponding to U_1 , by comparing the joint probability of interaction $P(\varepsilon)$ between tracks in ε , to the joint probabilities of interactions $P(\varepsilon')$, of all its respective subsets³ of tracks $\varepsilon' \subseteq \varepsilon$.

² Note that this does not preclude any of the objects involved in the candidate event graph being involved in other interactions during \mathcal{I} - e.g. answering the telephone while cooking a meal.

³

When ε is a set of two outcomes $\{x, y\}$, we have the well known form $PMI(x, y) = \log(P(x, y)P(x)^{-1}P(y)^{-1})$. This form is generalized to more than 2 variables in equation 1. Note that joint probabilities of subsets $\varepsilon' \subseteq \varepsilon$ with odd cardinality (e.g. $P(x), P(x, y, z)$) are in the denominator

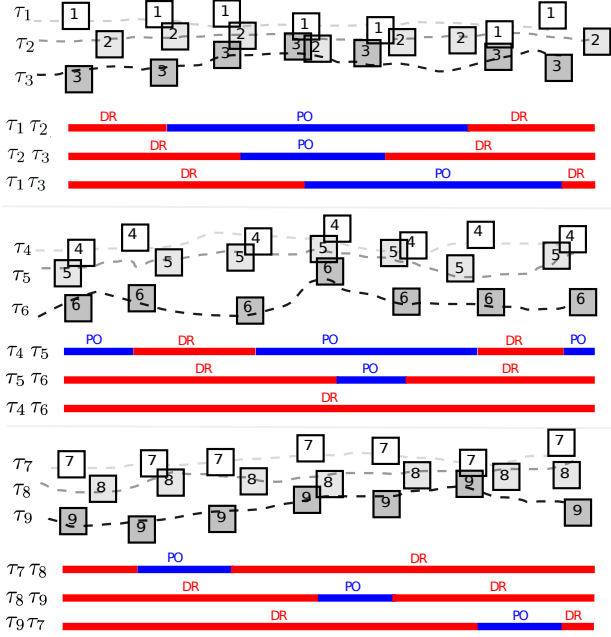


Figure 4. (a) Interactions between all three tracks τ_1, τ_2, τ_3 are uniformly distributed between all subsets of tracks and over the temporal period. (b) Interactions between tracks τ_4, τ_5 are far more than between the other subsets $\{\tau_5, \tau_6\}$ and $\{\tau_4, \tau_6\}$. (c) While interactions are evenly distributed between subsets of tracks, they are less evenly distributed temporally. Here, τ_7, τ_8 interact initially, while τ_9 is a *bystander*, and then τ_7, τ_9 interact while τ_8 is a *bystander*.

$$PMI(\varepsilon) := \log \left(\prod_{\varepsilon' : \varepsilon' \subseteq \varepsilon} P(\varepsilon')^{q_{\varepsilon'}} \right) \text{ where } q_{\varepsilon'} = (-1)^{\|\varepsilon'\|} \quad (1)$$

We adopt a well known procedure for estimating the joint probabilities $P(\varepsilon')$ in equation 1, by measuring the proportion of contexts (which is appropriately defined below) in which the interaction between all tracks in the subset ε' are observed, to the total number (N) of all possible contexts. We have found that a window of width w that captures w consecutive interactions of the activity is an appropriate context for our purpose⁴.

We formulate the window in terms of the activity graph by first noting that the layer 3 nodes labelled by *meets* capture (points of) interactions between all pair of tracks for the entire activity. We order these nodes temporally (by the end of each initial episode of the *meets* relation) to get a sequence $A = (\alpha_1, \dots, \alpha_N)$ as shown in Fig 5, where N is the total number of interactions for an entire activity. A window of width w is simply defined as a subsequence $(\alpha_k, \dots, \alpha_{k+w-1})$ of length w .

The probability of interaction $P(\tau_i)$ for a single track $\tau_i \in \varepsilon$, with respect to a candidate event graph H , is just the fraction of the total number of windows $N - w + 1$, in which τ_i interacts with any other track in U_1 . We estimate $P(\tau_i)$ from the activity graph, by counting the number windows ω , which contain layer 3 nodes labelled by *meets* in H , such that its descendants in layer 1 contain τ_i

since $q_{\varepsilon'} = -1$, while those of even cardinality (e.g. $P(x, y)$) are in the numerator, since $q_{\varepsilon'} = 1$.

⁴ A sliding *window* of a fixed width w (e.g. a window of w words) has been regarded as a good context in the statistical natural language processing community, where it is used to compute the association between co-occurring words (e.g. *bread and butter*) by computing their co-occurrence within such windows.

and all the layer 2 nodes are in H , and then normalizing by $N - w + 1$.

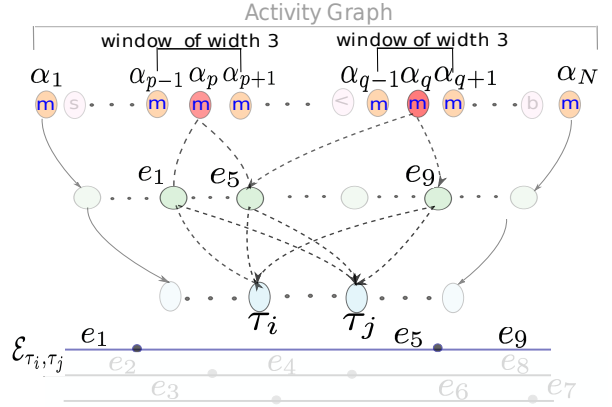


Figure 5. Windows on a sequence of layer 3 *meets* nodes $\alpha_1, \dots, \alpha_N$ of an activity graph are used to measure the joint probabilities for a set of tracks - here just τ_i and τ_j .

In a similar manner, the joint probability $P(\varepsilon)$ for any set of tracks ε , with respect to a candidate event graph H , is estimated by counting the number of windows ω of width w , which contain layer 3 *meets* nodes in H , the set of all layer 1 descendants of which are equal to ε and all the descendants in layer 2 are in H , and then finally normalizing by $N - w + 1$. Let $Desc_1(\bar{\alpha})$ and $Desc_2(\bar{\alpha})$ be the set of all descendants of $\bar{\alpha} \in A$ in layer 1 and layer 2 respectively of the activity graph G . We define the probability $P(\varepsilon) = \frac{\|\beta\|}{N - w + 1}$, where β as follows.

$$\beta := \{\bar{\alpha} = \omega \cap U^3 \neq \emptyset \wedge Desc_2(\bar{\alpha}) \subseteq U^2 \wedge Desc_1(\bar{\alpha}) \in U^1\}$$

In Fig. 5, the leftmost sample window of width 3 captures the interaction *meets*(e_1, e_5) between τ_i, τ_j as shown below the activity graph, as represented by the layer 3 node α_p . Similarly, the rightmost window captures the interaction *meets*(e_5, e_9) between τ_i, τ_j as shown below the activity graph, and as represented by the layer 3 node α_q . All the descendants of α_p and α_q in layer 1 are equal to $\{\tau_i, \tau_j\}$.

We insert the probabilities of interaction for all subsets $\varepsilon' \subseteq \varepsilon$ in equation 1 to obtain $PMI(\varepsilon)$, and thus measure the PMI for all subsets ε that correspond to the respective layer 1 nodes (tracks) for the candidate event graph H . The PMI for scenarios such as shown in Fig. 4 (a), in which interactions are temporally well interleaved, tend to be higher than scenarios such as Figs. 4 (b),(c), since the former kind are likely to induce more windows, for larger subsets of tracks (and therefore a greater PMI score).

We now compute pointwise total correlation $\xi(H)$ [16], which is the sum of all the pointwise mutual information $PMI(\varepsilon)$ over all subsets ε of tracks H_1 of a candidate event, weighted by their respective joint probabilities $P(\varepsilon)$.

$$\xi(H) := \sum_{\varepsilon : \varepsilon \subseteq H_1} P(\varepsilon) PMI(\varepsilon)$$

The value of $\xi(H)$ is highest when interactions between the tracks for the candidate event graph H are well distributed, both temporally and amongst subsets of tracks. Therefore we regard $\xi(H)$ as a good measure of *interactivity*. We define a candidate event graph H as being *sufficiently interactive* if $\xi(H)$ is greater than a threshold λ_2 .

4.2 Mining an Event Taxonomy

We apply a level-wise mining procedure [2] to obtain a hierarchy of event classes, where level i contains event classes with i layer 1 nodes. Initially $i = 2$, and the activity graph is searched to find *frequent* candidate event graphs which are *sufficiently interactive*. Instead of a threshold on frequency and interactivity, the top $k\%$ of frequent and interactive event classes are regarded as the initial set of event classes with 2 nodes in layer 1. The resulting event class graphs form a partonomy ordered by a part relation between sets of layer 1 nodes, as indicated in the *vertical dimension* of figure 6.

Level 3 event class graphs are found in the same way, by extending the event class graphs from level 2 and so on for each successive level until no maximally frequent and sufficiently interactive event class graphs can be found at the new level. By using the results of level i to form level $i + 1$, efficiency is improved over a search for level $i + 1$ graphs *de novo*.

However there is a second way in which one event class graph can be a subevent of another: the number of layer 1 nodes is identical, but the superclass contains more interactions. This aspect of the hierarchy is indicated in the *horizontal dimension* within each plane in Fig. ???. In both cases the subclass is a subgraph of the superclass.

Once the level-wise search terminates, all event classes which are *non-maximal* are eliminated, i.e. those classes all of whose instances appear as instances of a subgraph of its respective superclass. We also eliminate those graphs from the remaining event classes, which are not a subgraph of any graph of the respective superclasses. This way, we obtain event classes all of whose graphs naturally compose their respective superclasses and thus contribute to the entire structure of the taxonomy.

5 Experiments

5.1 Synthetic Data.

We perform experiments with synthetic data where (i) we artificially generate event classes; (ii) simulate tracks that interact according to relations given by these event classes (iii) add noise in these interactions; (iv) try to re-discover these original event classes using the proposed technique by varying the thresholds. These experiments suggested that retaining the top 30% of graphs at each level recovers most of the interesting patterns.

5.2 Evaluation on Real Data.

Experimental Setup. We evaluated the proposed method on approximately 12 hours of video showing servicing of aircraft between flights. There are eight turnarounds consisting of several classes of events such as unloading of luggage and bridge attaching to the plane etc. This dataset was chosen since it clearly contains structured events and these may occur in parallel with objects shared between events (e.g. the plane). The camera positioning for all the eight turnarounds is the same, so we obtain the same view. We first learned models for six visual appearance based object classes (1.Plane 2.Trolley 3.Carriage 4.Loader 5.Bridge 6.Plane Puller) from two turnarounds and then use the tracking techniques from [18] to generate tracks T for the other 6 turnarounds. Note that although the tracked objects have types as a result of the tracking technique we use, the event learning procedure deliberately ignores these in order not to be dependent on them. In principle, it could work equally with untyped tracks.

The Activity Graph We applied the proposed event learning framework to the tracks obtained from the test data of 6 turnarounds of aircraft handling videos. Using the tracks, we obtained the corresponding activity graph which consists of 749700 nodes.

The Event Taxonomy We applied the proposed level wise mining technique to the activity graph and obtained an event taxonomy shown in Fig. 6. The taxonomy represents a hierarchy of event classes, where level i contains event classes (shown as boxes) that represent spatio-temporal relations between i layer 1 nodes (tracks). Within each level i of the taxonomy, events are further organized as a hierarchy, where each sub-level j contain j *meets* relations, that is to say that they represent j *interactions* between the i tracks. A pair of numbers (i, j) indicating i tracks and j interactions are shown along side each level of the taxonomy. The part-of relations between the event classes are shown using connecting lines which span both *within* and *across* the levels of the event taxonomy.

In order to visualize the events in taxonomy, we have sampled some of the event classes marked using letters a to j , an event graph from each of these classes, and then displayed the corresponding interactions which are also marked with the respective letters a to j , in Fig. 6.

Qualitative Evaluation by Inspection. From an examination of the taxonomy and the sample interactions shown in Fig. 6, we can make the following observations. To start with, the taxonomy has been able to capture very simple interactions, such as between just two tracks given by the letters a and b , and their combination (attach-detach) in c , as captured by level 2 of the taxonomy. At the next level 3, interactions between trolley,plane and loader in d,e represent events that are typical for aircraft handling scenarios, as these interactions are central to the loading operation. The event indicated by f typically happens in the beginning of the turnover, when the bridge attaches itself to the plane, followed by the loader attaching itself to the plane, before the loading operations commence. The event given by letter j usually takes place in the middle of a turnover when multiple trollies arrive and depart with baggage. Finally, while the event given by h spans an entire turnaround, g,i take place towards the end as the plane puller attaches to the plane and the bridge detaches from the plane. It can also be seen that the taxonomy in Fig. 6 captures both within-level relations, for example e and d are a part-of h , and across-level relations, for example d is a part-of i .

From an inspection, as described above, we conclude that the proposed technique has discovered intuitive events in a natural taxonomic relation, that represent *commonly occurring* and *significant* interactions that take place in aircraft handling scenarios.

Quantitative Evaluation with a Pre-defined Set of Events We evaluate the performance of our event mining framework with respect to a pre-determined set of event classes - 1.Unloading 2.Bridge attaches and detaches from the plane 3.Plane Puller(PP) attaches to the plane. These classes were predefined as *interesting* with respect to monitoring tasks that were prescribed by domain experts. A ground truth of the 6 turnarounds for these three classes was defined by domain experts.

The proposed technique was able to discover (i) 51% of the unloading occurrences without and 73% with the HMM developed in our previous work [9] for obtaining robust qualitative spatial relationships; (ii) 66% of the Bridge attach-detach occurrences without the HMM and 83.3% with the HMM; (iii) 83.3% of the plane puller attach occurrences without the HMM and 100% with the HMM;

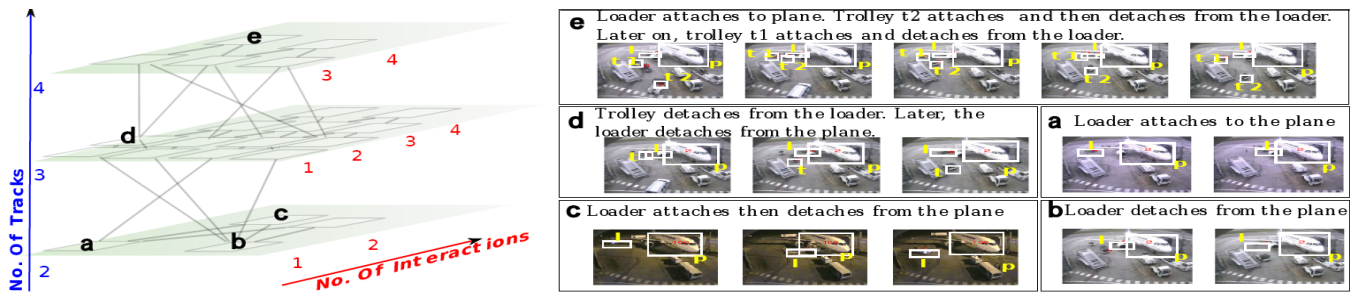


Figure 6. Sample events *a–e* from the taxonomy on the left are shown on the right. In the image sequences on the right, next to the respective bounding boxes are symbols that stand for the object types : b - bridge, p - plane, pp - plane puller, t - trolley, l - loader.

We conclude that despite the imperfect inputs from tracking and absence of any supervision for each of these event classes, the proposed technique has discovered these pre-defined events with a reasonably high accuracy. We also conclude that the proposed HMM based technique for obtaining qualitative spatial relations from potentially noisy tracking output offers a robust alternative for using qualitative spatio-temporal relations to discover events.

6 Summary and Future Work

In this work we have proposed a framework for unsupervised discovery of event class taxonomy from videos with the following features: (i) activity graph – a variable free graph based representation of spatio-temporal relationships for an activity; (ii) interactivity and frequency – two criteria to steer the search for subgraphs of the activity graph which are event-like; (iii) a level wise procedure for automatically constructing an event taxonomy; Experiments to evaluate the proposed technique have shown that the event taxonomy represents intuitive events in a natural taxonomic relations on real video data which are complex due to multiple overlapping events that may share participating objects.

In the future, we plan to use the learned event class taxonomy in several ways. First, the learned classes may be used for detecting events in an unseen video or classifying unseen events as normal or abnormal. We also plan to experiment with techniques described in [9] showing how to learn functional object classes. Further challenges include combining object appearance features as a part of event definitions.

REFERENCES

- [1] Laura-Andreea Antanas, Ingo Thon, Martijn van Otterlo, Niels Landwehr, and Luc De Raedt, ‘Probabilistic logical sequence learning for video’, in *Preliminary Proceedings of the International Conference on Inductive Logic Programming (ILP-2009)*, (July 2009).
- [2] D J Cook and L B Holder, *Mining Graph Data*, Wiley-Interscience, 2007.
- [3] Raffay Hamid, Siddhartha Maddi, Amos Johnson, Aaron Bobick, Irfan Essa, and Charles Isbell, ‘A novel sequence representation for unsupervised analysis of human activities’, *Art. Int. Journal*, (2009).
- [4] L. B. Holder, D. J. Cook, and S. Djoko, ‘Substructure discovery in the subdue system’, in *Proc. of the AAAI Workshop on Knowledge Discovery in Databases*, pp. 169–180, (1994).
- [5] A Inokuchi, T Washio, and H Motoda, ‘An apriori-based algorithm for mining frequent substructures from graph data’, in *PKDD ’00: Proc. 4th Eur. Conf. on Principles of Data Mining and Knowledge Discovery*, pp. 13–23, London, UK, (2000). Springer-Verlag.
- [6] J. Kabler, U. Schaning, and J. Toran, ‘The graph isomorphism problem: Its structural complexity’, *Birkhauser*, (1993).
- [7] Kris M. Kitani, Yoichi Sato, and Akihiro Sugimoto, ‘Recovering the basic structure of human activities from noisy video-based symbol strings’, *IJPRAI*, **22**(8), 1621–1646, (2008).
- [8] M Kuramochi and G Karypis, ‘Frequent subgraph discovery’, in *ICDM*, pp. 313–320, (2001).
- [9] Muralikrishna Sridhar, Anthony G. Cohn, and David C. Hogg, ‘Learning functional object-categories from a relational spatio-temporal representation’, in *Proc. ECAI 2008*, pp. 606–610, Amsterdam, The Netherlands, The Netherlands, (2008). IOS Press.
- [10] Muralikrishna Sridhar, Anthony G. Cohn, and David C. Hogg, ‘Discovering an event taxonomy from video using qualitative spatio-temporal graphs’, in *Proc. ECAI 2010*. IOS Press, (2010).
- [11] Muralikrishna Sridhar, Anthony G. Cohn, and David C. Hogg, ‘Unsupervised learning of event classes from video’, *Proc. AAAI*, (2010).
- [12] C Wang, Y Zhu, T Wu, W Wang, and B Shi, ‘Constraint-based graph mining in large database’.
- [13] Xiaogang Wang, Xiaoxu Ma, and Grimson, ‘Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models’, *IEEE TPAMI*, **31**(3), 539–555, (2009).
- [14] Xiaogang Wang, Kinh Tieu, and Eric Grimson, ‘Learning semantic scene models by trajectory analysis’, in *In ECCV (3) (2006)*, pp. 110–123, (2006).
- [15] Takashi Washio and Hiroshi Motoda, ‘State of the art of graph-based data mining’, *SIGKDD Explorations*, **5**(1), 59–68, (2003).
- [16] S. Watanabe, ‘Information theoretical analysis of multivariate correlation’, *IBM Journal of Research and Development*, **4**, 66+, (1960).
- [17] Xifeng Yan and Jiawei Han, ‘gspan: Graph-based substructure pattern mining’, in *ICDM ’02: Proceedings of the 2002 IEEE International Conference on Data Mining*, p. 721, Washington, DC, USA, (2002). IEEE Computer Society.
- [18] Qian Yu and Gerard Medioni, ‘Integrated detection and tracking for multiple moving objects using data-driven mcmc data association’, *Motion and Video Computing, IEEE Workshop on*, **0**, 1–8, (2008).
- [19] Feida Zhu, Xifeng Yan, Jiawei Han, and Philip S. Yu, ‘gprune: A constraint pushing framework for graph pattern mining’, in *PAKDD*, eds., Zhi-Hua Zhou, Hang Li, and Qiang Yang, volume 4426 of *Lecture Notes in Computer Science*, pp. 388–400. Springer, (2007).