Event Detection in Complex Scenes Using Interval Temporal Constraints

Yifan Zhang[1], Qiang Ji[2] and Hanqing Lu[1]
[1]NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2]Rensselaer Polytechnic Institute, Troy, NY 12180, USA
{yfzhang, luhq}@nlpr.ia.ac.cn, qji@ecse.rpi.edu

Abstract

In complex scenes with multiple atomic events happening sequentially or in parallel, detecting each individual event separately may not always obtain robust and reliable result. It is essential to detect them in a holistic way which incorporates the causality and temporal dependency among them to compensate the limitation of current computer vision techniques. In this paper, we propose an interval temporal constrained dynamic Bayesian network to extend Allen's interval algebra network (IAN) [2] from a deterministic static model to a probabilistic dynamic system, which can not only capture the complex interval temporal relationships, but also model the evolution dynamics and handle the uncertainty from the noisy visual observation. In the model, the topology of the IAN on each time slice and the interlinks between the time slices are discovered by an advanced structure learning method. The duration of the event and the unsynchronized time lags between two correlated event intervals are captured by a duration model, so that we can better determine the temporal boundary of the event. Empirical results on two real world datasets show the power of the proposed interval temporal constrained model.

1. Introduction

Event detection plays an essential role in video content analysis and has received increasing attention from computer vision researchers for decades. However, the study on detecting events in complex scenes with multiple persons/objects either in interaction or as a group is still limited. In a complex scene, multiple events often occur sequentially or in parallel over a period of time. Each event may be correlated and affected by others. Detecting each individual event separately may not always obtain reliable result due to many reasons such as occlusion, motion blur, appearance variation, background clutter, *etc*. It is essential to detect them in a holistic way which incorporates the causality and temporal dependency among them to compensate the limitation of current computer vision techniques.

In a complex scene, the atomic events often maintain certain temporal relationships with each other, and their occurrences are governed by an underlying temporal structure based on some domain knowledge and rules of thumb. For example, in a basketball game, one should *catch* the ball after another one *pass* it. The action *shooting* should be finished during *jumping*. In a traffic junction, the straight traffic sometimes temporally overlaps with the turning traffic. If we can tell how long they overlap, and in what delay the later one still lasts after the early one has finished, that would be great helpful to detect the event and better determine the event boundary. Hence, it is important to capture the interval based temporal relationships and discover the underlying temporal structure amongst the events, which can be used as an inference engine to disambiguate the uncertainties from the low-level visual processing and facilitate the event detection.

For various methodologies which can model multiple event relationships and interactions, such as graphical, description-based and logic-based models [16], they may face one or more of the following issues:

(1) Most of the methods typically assume the events occurring instantaneously, which is unrealistic for many real world applications. Thus they can only offer three time-point relationships (*i.e*. before, after and equal), and are not expressive enough to capture a larger number of interval based temporal relationships such as during, overlapping, *etc*. between the events.

(2) Some explicit duration graphical models such as semi-HMM [7], determine the event duration solely by the event itself, regardless of the impacts from the others, as such, they cannot capture the unsynchronized temporal lags between the event intervals and lack the ability to fully express the interval temporal relationships.

(3) Most of the multi-thread models do not perform structure learning, with the model structure manually specified (*e.g*. description-based methods [1]) or fully connected (*e.g*. Coupled HMM [12]), which may either suffer when the domain knowledge is unknown or risk expensive computational cost when modeling more parallel events.

To address these issues, we develop an interval temporal constrained dynamic Bayesian network to extend Allen's interval algebra network (IAN) [2] from a deterministic static model to a probabilistic dynamic system. The structure of the model is optimized by an two-stage structure learning method. In the first stage, the IAN is constructed by analyzing the interval temporal relationships among the events, and its topology is used as the prior structure within each time slice. In the second stage a score searching based learning algorithm is performed to refine the prior structure and discover the salient interlinks between the adjacent time slices. Moreover, we do not simply model the events as occurring instantaneously. A duration model is attached to the DBN to capture the interval length of the event. Different from the existed explicit duration models, in which the duration is only determined by the event itself, we claim that, the duration of the dependent event is also affected by the status of the depended one, given their interval temporal relationships. Thus, a duration fragmentation is performed to better represent the interval temporal relationships, and thus we can determine the start and end point of the event more accurately.

1.1. Related work

Among various methodologies which can model multiple event relationships and interactions, time-sliced graphical models, i.e. hidden Markov models (HMMs) and dynamic Bayesian networks (DBNs), have become the most popular tool for modeling and detecting visual events. Oliver *et al*. [12] exploited coupled hidden Markov models (CHMMs) to model basic human interactions such as one person following another, altering their path to meet another and so forth. Xiang and Gong [18] presented a dynamically multi-linked hidden Markov model (DML-HMM) for modeling the temporal and causal correlations among events in an outdoor scene. Pinhanez [13] captured relative temporal relationships in a propagation network to detect event in a deterministic way, which cannot handle the uncertainties brought by the visual observation. Generally, time-sliced graphical models typically model events as occurring instantaneously, so as to lack the expressive power to capture a fully range of the interval temporal relationships.

To explicitly model the duration of the event, Hongeng and Nevatia [7] made use of semi-HMM to relax the Markovian assumption. Natarajan and Nevatia [11] then coupled multiple chains of semi-HMM and proposed coupled hidden semi-Markov model (CHSMM) to model interactions among temporal entities. Shi *et al*. [15] introduced a DBN framework that provides duration modeling within the network with the limited ability to capture only simple sequential temporal relationships such as before or after. Duong *et al*. [6] proposed a switching hidden semi-Markov model (S-HSMM) for recognizing a sequence of events. However, it cannot handle the scenarios with multiple parallel streams of events. Most of the explicit duration models determine the event duration solely by the event itself, while discarding the implied affection from the other depended events, thus cannot well capture the unsynchronized temporal lags between the correlated event intervals. Moreover, these methods seldom perform structure learning process, leaving the model structure manually defined or fully connected, which cannot automatically discover the undying temporal constraints beneath the observations.

Logic-based and topic-based approaches have also gained attention in recent years for solving visual modeling problems. Morariu and Davis [10] proposed an Markov logic network based approach for complex multi-agent event recognition that employs knowledge such as rules, event descriptions, and physical constraints of the events being modeled. Probabilistic event logic (PEL) proposed by Brendel *et al*. [4] is a probabilistic treatment of EL based on confidence-weighted formulas, similar as MLN is to first-order logic. However, both of them primarily specify the model structure and parameters manually. The related rules and relations must be known in advance to encode them into the logic formulas. Kuettel *et al*. [9] proposed a DDP-HMM model to recognize the activities in traffic scenes, in which each event corresponds to a topic which is a specific spatial flow pattern. Varadarajan *et al*. [17] used a topic model to capture global and local rules of surveillance scenes within a probabilistic generative process, in which the relationships among events are limited to simple relationships such as before, after or equal. The topic models, while powerful in modeling some complex activities, still cannot effectively handle events with strong and diverse temporal dependencies.

Based on the limitations of the approaches mentioned above, we need to find a model which can systematically accounts for a full range of interval temporal relationships among different events. The relationships should be automatically discovered. Moreover, it should be a probabilistic model which can handle the uncertainty brought by the low-level visual processing.

2. Interval Algebra Network (IAN)

An event is defined as the state change of one or more entities over a period of time. Events occur over intervals of time and are correlated by their temporal relationships. According to Allen's axiomatization of time periods [2], there are thirteen atomic relations $\{b, bi, m, mi, o, oi, s, si, d, di, f, fi, eq\}$ that can hold between two events, and they respectively represent, as shown in Fig. 1, before, meets, overlaps, starts, during, finishes, equal, and their inverses. The actual interval relationship between two events that happens over a time interval can be a union of these atomic relations, e.g., $Y\{b, m\}X$ repre-

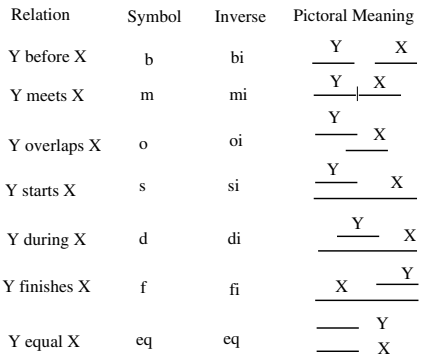

Figure 1. Allen's thirteen atomic interval temporal relations to represent the temporal relations between two events X and Y.

	r_P^t	r_N^t	r_F^t
e	P	N	F
b	PNF	F	F
ib	P	P	PNF
m	PN	F	F
im	P	P	NF
o	PN	NF	F
io	P	PN	NF
s	PN	N	F
is	P	PN	F
d	PN	N	NF
id	P	PNF	F
f	P	N	NF
if	P	NF	F

(a)

	r_T^t	r_F^t
e	T	F
b	F	TF
ib	F	TF
m	F	TF
im	F	TF
o	TF	TF
io	TF	TF
s	T	TF
is	TF	F
d	T	TF
id	TF	F
f	T	TF
if	TF	F

(b)

Table 1. Mapping the 13 interval temporal relations into intra-slice pairwise constraints using the 3-valued state domain (a), where the symbols "P","N" and "F" represent "past", "now" and "future"; and the 2-valued state domain (b), where the symbols "T" and "F" represent "true" and "false".

senting $(Y\, before\, X)$ or $(Y\, meets\, X)$. An interval algebra network [2], or simply IAN can be used to represent the temporal relationships among a set of events, where the nodes represent events, and the directed links represent the temporal relationships among the events. Each link is labeled with the union of all possible interval relations between the two events. Fig. 3 shows an IAN that models the interval temporal relationships among 6 events in basketball games. For the sake of readability, the node which the link heading to is defined as the "temporal dependent", and the node which the link coming from is defined as the "temporal reference".

3. IAN representation in a probabilistic dynamic model

Despite the capability of representing the temporal structure among time intervals, IAN is a deterministic static model, which cannot model the temporal evolution of the events, and is not able to handle uncertainties from the low-level visual processing. Hence, we propose to project the IAN into a probabilistic dynamic system, thus forming an Interval Temporal constrained Dynamic Bayesian Network. A DBN is a directed acyclic graphical model, which models the temporal evolution of a set of random variables X over time. It is defined as $B = (G, \Theta)$, where G is the model structure, i.e., the nodes and the links, and Θ represents the model parameters, i.e., the Conditional Probability Distributions (CPDs) for all nodes.

Usually, a time-sliced model can not effectively handle relationships occurring over time intervals. To fully represent the total thirteen Allen's interval temporal relations in a time-sliced model, we share the idea from Pinhanez and Bobick's work [13]. In a DBN, we extend the state domain of the event node from a traditional 2-valued domain *m* = {true(T), false(F)} to a new 3-valued domain *m* = {past(P), now(N), future(F)}, which means "the event has finished already", "it is happening now" and "it will happen in the future", respectively. By using this 3-valued state domain, the 13 interval temporal relations are systematically transformed into pairwise temporal constraints to restrict the admissible states of dependent event node given its temporal reference.

Intra-slice constraint: Generally, the pairwise constraints carried on intra-slice links in a DBN represent the causal relationship. However, by using the 3-valued state domain, they can also reflect the temporal relationship. For instance, suppose e_i *meets* e_j. Setting e_i as the temporal reference, if e_i is happening now, notated as $e_i^t = N$, then e_j can only occur in the future, notated as $e_j^t = F$. Similarly, if $e_i^t = F$, then $e_j^t = F$. If $e_i^t = P$, then $e_j^t = PN$, meaning that the event e_j is happening now or have finished already. Table 1 (a) displays the mapping from the 13 interval temporal relations to the equivalent intra-slice pairwise constraints represented by "P/N/F" value, where r_P^t represents the admissible values of e_j^t given its temporal reference $e_i^t = P$, and similarly for r_N^t and r_F^t. For comparison, the mapping results to the "T/F" value domain is shown in Table 1 (b). It can be seen that using the 3-valued state domain is more expressive to represent the 13 interval temporal relations than using the 2-valued state domain.

Inter-slice constraint: In [13], the interval temporal relations are only mapped to the intra-slice pairwise constraints. However, in many cases, the temporal dependent e_j is not only restricted by the current state of its temporal reference e_i at time t but also the previous state of e_i at time $t-1$. For instance, suppose e_i *overlaps* e_j. If $e_i^t = P$ and $e_i^{t-1} = P$, then $e_j^t = PN$, meaning that e_j may be happening now or have already finished. However, if $e_i^t = P$ and $e_i^{t-1} = N$, indicating that e_i just finished at time $t-1$, then e_j must be happening now, that is $e_j^t = N$. Hence, the inter-slice constraints from the temporal reference events at the previous time slice are also critical to reveal the interval temporal relations. They can be considered as inter-slice pairwise constraints, and are represented as the inter-slice links from the correlated event nodes in

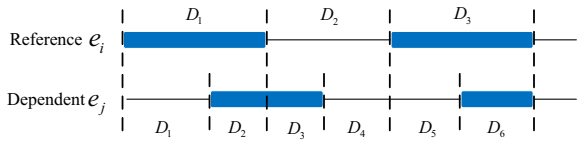

Figure 2. The duration of the dependent event e_j is fragmented by the state transition point of its reference event e_j.

a DBN model. Besides the inter-slice pairwise constraint, the evolution of each event is also restricted by the previous state of itself, which can be called inter-slice self constraint. Given the previous state, the event node can be either stay in the same state or transit to a restricted state. That is, $\mathcal{T}(P) = F$,$\mathcal{T}(N) = P$,$\mathcal{T}(F) = N$, where $\mathcal{T}(\cdot)$ means state transition. Particularly, the state "past" jump to "future" means that after a certain time when the event had finished, it will revisit the state of waiting for the new occurrence.

Interval duration: To capture the duration of the event and better represent the interval temporal relations, each event node is attached with a duration node in the model. Different from most of the existed explicit duration models, in which the duration is only determined by the event itself, the duration node of the dependent event in our model is also conditioned on the status of its temporal reference, given their interval temporal relationships. For example, if e_i *starts* e_j, the duration conditioned on $e_i = P$ and $e_j = N$ is the length of the interval in which e_j still lasts while e_i has already finished. Also, if e_i is *before* e_j, the duration conditioned on $e_i = P$ and $e_j = F$ can tell in what delay e_j will occur after e_i has already finished. Since the duration node of the dependent event has multiple parents, it actually performs a duration fragmentation which is shown in Fig 2. The interval of the dependent event being in the same state is fragmented by the time point of the reference event state transition. By this, we can provide a quantitative description for the unsynchronized time lags between two events, and thus better model their interval temporal relations.

4. Structure learning

In a DBN model, both the intra-slice and inter-slice constraints are embedded in the links. We want to learn the temporal and causal correlations by finding a DBN structure that can best explain the observation in the training data. The structure of the model is discovered by an two-stage structure learning method.

4.1. Prior structure construction

In structure learning algorithm, a sophisticate structure initialization method is required to get a good starting point to avoid getting stuck at a local maximum during structure space searching. Therefore, we construct an IAN for the events by analyzing the interval temporal relationships between them in the training data. The topology of the IAN can be used as the prior structure of the BN on each time slice of the DBN. To construct the IAN, we use a temporal window with predefined length sliding along the time axis in the training data to get temporal interval samples. Thus we can obtain the statistics of the temporal relationships for each pair of events by analyzing every sampled temporal interval. The pairwise temporal dependency between event e_j and its temporal reference e_i is represented by $P(e_j = 1|_r e_i = 1)$, that is the probability of e_j being present and related with e_i by temporal relation r conditioning on e_i being present. We call it the "r related co-occurrence conditional probability". It is computed as follows:

$$P(e_j = 1|_r e_i = 1) = \frac{N_{e_i \wedge_r e_j}}{N_{e_i}}, \tag{1}$$

where $N_{e_i \wedge_r e_j}$ is the total number of "r related co-occurrences" of e_i and e_j in the sampled temporal intervals regardless of the presence of other events, and N_{e_i} is the total number of occurrence of e_i. Please note that the dependency between two events is not symmetric, that is, $P(e_j = 1|_r e_i = 1) \neq P(e_i = 1|_r e_j = 1)$. Due to the asymmetry, we use a directed graph instead of an undirected graph to represent the IAN.

For each event e_i and its temporal reference e_j, we get the maximal r related co-occurrence conditional probability, $\tilde{P}_{ij} = \max_r P(e_j = 1|_r e_i = 1)$. If $\tilde{P}_{ij}$ is higher than a predefined threshold, we assume that e_i has a strong temporal dependency with e_j, which can be modeled with a link from e_i to e_j. In the IAN, the links are added in a sequential manner from the largest value to the smallest value of $\tilde{P}_{ij}$ for each i and j. Algorithm 1 is the IAN construction algorithm.In this algorithm, the newly added links should follow both the *DAG* consistency and the temporal consistency(*TC*). The *DAG* consistency makes sure the graph shall be a directed acyclic graph. The *TC* makes sure that the temporal relationship on the newly added link must be consistent with the temporal relationship on the existed links. Specifically, if a link is added in the current graph and form a triangle with two existed links, the temporal relationship on the new link should satisfy the transitivity rules governed by the temporal relationship on the two existed links. The lookup table of the transitivity rules please refer to Fig. 4 in [2]. Fig. 3 shows a constructed IAN modeling the interval temporal relationships among 6 events from the OSUPEL basketball data [4]. After the IAN is obtained, its topology is directly used as the prior structure of the BN on each time slice of the DBN, while the inter-slice links are initialized corresponding to the intra-slice links accordingly.

4.2. Structure optimization

After constructing the IAN, we obtain an initial DBN structure. Although it is our best guess based on the temporal relationship analysis, it may not be correct enough to

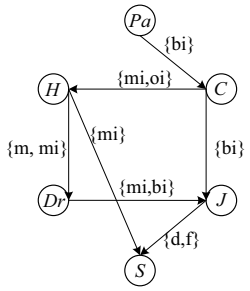

Figure 3. The basketball IAN modeling the interval temporal relationships among six events from the OSUPEL basketball data. The names of the events are abbreviated to: Pa=Pass, C=Catch, H=Hold ball, Dr=Dribble, J=Jump, S=Shot.

Algorithm 1 IAN construction algorithm

Input: The set C of all the maximal r related co-occurrence conditional probabilities for each event e_i and e_j: $\{\tilde{P}_{ij}\}_{i,j=1,...,K;i\neq j}$.
Output: The constructed IAN G_n;

```
Initialize the IAN structure to G_0 without any link between nodes, where one node corresponds to one event;
while Set C ≠ ∅ do
    find the maximum value P̃_{i*j*} in Set C;
    if P̃_{i*j*} > Th then
        Get G_{n+1} by adding link from i* to j* in G_n;
        if G_{n+1} does not satisfy DAG and TC then
            G_{n+1} = G_n;
        end if
    end if
    Delete P̃_{i*j*} from Set C;
end while
return G_n
```

reflect the true relationships. Given a set of observed data $\{D_1, D_2, ..., D_M\}$, where M is the total number of the video frames, we can refine the initial DBN model with a structure learning algorithm, i.e., finding a DBN structure G that best fits the observed data.

The structure learning algorithm first defines a score that describes the fitness of each possible structure G to the observed data, and then, the best fitted network structure is identified with the highest score. The fitness score is defined as

$$Score(G) = logP(D, G) = logP(G) + logP(D|G), \quad (2)$$

where $logP(G)$ is the log prior probability of the DBN structure and $logP(D|G)$ is the log likelihood of the training data.

A DBN B can be defined as a pair (B_s, B_t): the static model $B_s = (G_s, \Theta_s)$ captures the static distribution over all variables X^0 in the first time slice, the transition model $B_t = (G_t, \Theta_t)$ captures the transition probability $P(X^{t+1}|X^t)$ for all t in finite time slices T. Hence, the fitness score is decomposed into two parts:

$$Score(G) = Score(G_s) + Score(G_t), \quad (3)$$

where $Score(G_s)$ and $Score(G_t)$ represent the score of the static network and the score of the transition network, respectively. Thus, we can learn the structure of G_s and the structure of G_t separately.

For the static model B_s, we first define the prior probability $P(G_s)$ for each structure. Instead of giving an equal prior $P(G_s)$ to all possible structures, we assign a high probability to the prior structure $G_{s_{prior}}$ which has the same topology of the constructed IAN. The prior probability of any other structure is decreased depending on the deviation to the prior structure as the way in [5]. The likelihood of the training data can be approximated by the Bayesian information criterion (BIC) [14] as follows:

$$logP(D_s|G_s) \approx logP(D_s|G_s, \hat{\Theta}_s) - \frac{dim_s}{2} log(L), \quad (4)$$

where $\hat{\Theta}_s$ is the set of parameters of G_s which maximizes the likelihood of the training data D_s. During model learning, the training data D with total M video frames is divided into L sequences with length m_l so that $\sum_{l=1}^{L} m_l = M$. D_s is the collection of all the first frames from every sequence to learn the static network structure G_s. L is the number of the first frames from all sequences, dim_s is the number of free parameters in G_s. In (4), the first term evaluates how well the model fits the data, and the second term is a penalty term to punish the structure complexity.

Given the definition of $Score(G_s)$, we employ an iterated hill climbing algorithm to search the optimal network structure. Starting from the prior static structure $G_{s_{prior}}$, we iteratively generate the nearest neighbor of $G_{s_{prior}}$ by adding, deleting or reversing a single link which is subject to the DAG constraint. The $Score(G_s)$ of the structure generated in each iteration is evaluated, and the one with the maximum score is selected as the structure of the static model.

In the transition model B_t, it contains both the intra-slice links and the inter-slice links. The prior intra-slice structure is set as the same topology of the IAN. The prior inter-slice structure is obtained by setting the links from the previous slice corresponding to the intra-slice links accordingly. Fig.4 (a) is a prior transition structure constructed based on the basketball IAN shown in Fig.3. Similar to the static structure learning, we assign a high probability to the prior transition structure $G_{t_{prior}}$. The likelihood of the training data given the transition structure is computed as follows:

$$logP(D_t|G_t) \approx logP(D_t|G_t, \hat{\Theta}_t) - \frac{dim_t}{2} log(M - L), \quad (5)$$

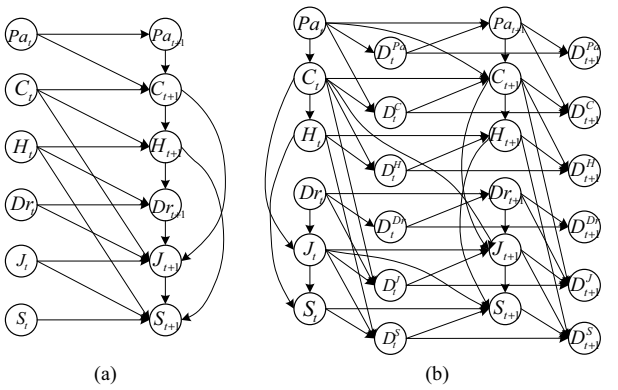

Figure 4. (a) The prior transition structure constructed based on the basketball IAN in Fig. 3; (b) the learned DBN model structure for OSUPEL basketball data. For clarity, the observation nodes of each event node are omitted.

where $\hat{\Theta}_t$ is the set of parameters of G_t which maximizes the likelihood of the training data D_t. D_t is the collection of data to learn the transition structure G_t. $M - L$ is the number of pairwise transitions between two adjacent slices from all training sequences, dim_t is the number of free parameters in G_t.

Given the definition of $Score(G_t)$, we apply the same iterated hill climbing algorithm to search the optimal network structure subject to some coherent constraints on the transition network. First, the nodes on the previous time slice, as shown in Fig.4 (a), do not have parents. Second, the inter-slice links can only direct from the previous time slice to the current time slice. Finally, based on the stationary assumption, both the inters-lice links and intra-slice links should be repeated for all time slices.

4.3. Duration and observation node

After we obtain the skeleton structure of our DBN model, each event node is attached with a duration node as its child. The state of the duration node represents how long the current state of the event node lasts. The duration node deterministically counts down on every time slice, and the event node state will not change until its duration node counts down to 0. Specially, the duration node of the temporal dependent event has the link also from the temporal reference event node.

Besides the duration node, each event node is also attached with an observation node as its child so as to form a two-layer model. The top layer encodes the events and their temporal relationships. The bottom layer comprises a set of observation nodes that ingest the preliminary detection from low-level features. The final learned DBN model with the duration nodes attached is shown in Fig.4 (b). Comparing to its prior transition structure, the learned structure has removed several unnecessary links within and between the time slice.

5. Parameter learning and inference

The parameters of the model are the Conditional Probability Distributions (CPDs) for all the nodes. The CPD of the event node e_k can be written as follows:

$$P(e_k^t{=}j|e_k^{t-1}{=}i,p_k^t{=}m,q_k^{t-1}{=}n,D_k^{t-1}{=}d){=}\begin{cases}\delta(i,j) & \text{if } d>0\\ A(m,n,i,j) & \text{if } d=0\end{cases} \tag{6}$$

where p_k is the configuration of intra-slice event parents of e_k, q_k is the configuration of inter-slice event parents of e_k, D_k is the duration node of e_k. $A(m,n,i,j)$ is the state transition probability given its event parents. When $d>0$, the state of e_k cannot be changed. Particularly, when $d=0$, e_k is not forced to transit to the next state. It can still stay at the same state because of the duration fragmentation described in section 3. The CPD of the duration node D_k is as follows:

$$P(D_k^t{=}d'|D_k^{t-1}{=}d,e_k^t{=}i,p_k^t{=}m){=}\begin{cases}\delta(d',d-1) & \text{if } d>0\\ \mu(i,m,d') & \text{if } d=0\end{cases} \tag{7}$$

where $\mu(i,m,d')$ follows a multinomial distribution. When parameter learning, since the training data can be fully observed, a Maximum Likelihood Estimation (MLE) is performed to learn all the CPDs given the complete data.

During model inference, the event nodes and the duration nodes in the top layer are hidden and need to be inferred from the observations in the bottom layer. The inference is conducted by finding the most probable explanation (MPE) of the observations. Let $e_{1:n}^t$ represents all the event nodes at time t, where n is the number of the event nodes. Given all the available observations until time T: $O_{e_{1:n}}^{1:T}$, the events nodes are inferred over time by maximizing the probability $p(e_{1:n}^t|O_{e_{1:n}}^{1:T})$. Since all the node in the model are discrete, the inference can be solved by a popular discrete DBN inference algorithm, known as B-K algorithm [3].

6. Experiments

In this section, we report the event detection results in complex scenes using the proposed model. Specifically, the results on two real datasets, the OSUPEL basketball data [4] and the QMUL Junction data [8], are discussed.

6.1. OSU basketball experiments

The OSUPEL basketball dataset is publicly available and it consists of multiple players playing against each other in a real basketball court. The videos show a real-world setting with the following challenges: frequent inter-player occlusions, camera motion, player's scale changing, motion blur *etc*. In the dataset we defined six types of events: Pass, Catch, Hold ball, Shoot, Jump and Dribble.

Before discussing the event detection results in the OSUPEL basketball dataset, we first briefly describe the method

on how to get the visual observation of the events from low-level features. The computed tracks of the players in the videos have been already provided in the dataset. We extract features from the bounding box of the tracks and use an HMM to detect each event separately. The features are derived from HoG and HoF. The HMMs are trained for each event class and used to detect events in the videos separately. Please note that the preliminary detection results are not satisfying, whose performance are summarized in the 1st row of Table 2 and 3. They are considered as noisy observations to feed into the bottom layer of our model so as to infer the state of the hidden nodes in the top layer.

To evaluate our model, the experiment was performed with a 5-fold cross validation setting, where the model was learned using 80% of the data and tested on the rest 20%. For comparing to the competing methods, we choose the coupled hidden semi-markov model (CHSMM) since it has been demonstrated in [11] that the CHSMM outperforms other HMM variants such as the CHMM, the HSMM and the S-HSMM. We also implemented the DML-HMM [18] which is a typical time-sliced model without duration nodes, whose structure is automatically learned. Similar to our model, both of these two methods can model the relationships between multiple events. In the experiment, we use the obtained preliminary detection results as the observations for each model to infer the true occurrence of the events. The F1-score of the event detection on both interval level and frame level are demonstrated in Table 2 and 3.

Comparing to the observations which are separately detected by the preliminary detectors, it is clear that the detection results are improved by detecting multiple events in a holistic way with using the relationships and constraints between events. In particular, our model performs better than CHSMM on both interval and frame level. It is proved that our structure learning method can discover the salient relationships and thus construct a more appropriate structure than the fully connected structure of CHSMM which contains a lot unnecessary links. To further verify the effectiveness of the IAN initialization for structure learning, we randomly initialized the DBN structure and construct the model. The results showed that the overall F1 score of event detection decreases by 7% and 10% on interval and frame level respectively. Our model is also superior to DML-HMM which does not explicitly model the event durations. Without duration model, it lacks the ability to fully express the interval temporal relations. In addition, the duration fragmentation is also an important step in our duration model. It can provide a quantitative description of the unsynchronized time lags between two intervals, thus can better determine the boundary of the event. Based on a comparison experiment, the overall F1 score of event detection on the frame level decreases by 4% without using duration fragmentation.

Table 2. Event detection performance on interval level

	Dribble	Jump	Shoot	Pass	Catch	Hold	Overall
Observation	0.65	0.43	0.27	0.37	0.35	0.56	0.47
CHSMM	0.68	0.46	**0.36**	0.49	0.51	0.70	0.57
DML-HMM	0.68	0.40	0.32	0.45	**0.54**	0.65	0.53
Our method	**0.71**	**0.50**	0.34	**0.55**	0.53	**0.71**	**0.61**

Table 3. Event detection performance on frame level

	Dribble	Jump	Shoot	Pass	Catch	Hold	Overall
Observation	0.51	0.45	0.25	0.31	0.33	0.47	0.43
CHSMM	0.61	0.48	**0.33**	0.42	0.49	0.54	0.54
DML-HMM	0.56	0.41	0.31	0.38	0.49	0.51	0.50
Our method	**0.64**	**0.51**	0.31	**0.46**	**0.51**	**0.62**	**0.58**

6.2. QMUL Junction experiment

QMUL Junction dataset contains a 60 minutes video which shows a busy traffic intersection where are three dominant traffic flows in different directions. By applying a topic model [9] with the feature of optical flow and position information, we can obtain several topics, among which we select 5 meaningful and salient topics to define them as the events: (A) vehicles moving from bottom to top, (B) from top to bottom, (C) from left to right, (D) from right to left, (E) from bottom and turning towards the right. These 5 events are regulated by the traffic lights and the right of way, thus they have strong interval temporal relationships.

The video is divided into a sequence of 3-second clips. For each clip we can get the distribution on the topics, so that we can determine which event occurs in the clip. Multiple events may co-occur in the same clip. Using this data of complex scene, our model can be learned to capture the interval temporal relationships and the event durations. The preliminary detections of the 5 events occurrence are quite accurate, which do not have much space to be refined. Hence, to evaluate the robustness of our model, the detections are corrupted by 2 types of noises which are common in event detection and fed to the model as the observations. We want to see whether our model is robust to the noises.

One common noise in event detection is mis-detection, i.e., the event is not detected or falsely recognized as another event. This experiment studies the performance of our model under a varying amount of mis-detection rate, i.e., 30% , 40%, and 50% events are mis-detected. This is accomplished by perturbing the event labels of the testing data to simulate incorrect event detection. Table 4 shows the performance of CHSMM, DML-HMM and our model under different mis-detection rates. As expected, the F1-score degrades when the mis-detection rate increases. However, the performance of our model remains higher than the other two models. It is relatively stable and decreases gradually as the mis-detection rate increases. This result shows that our model is more robust to event mis-detection compared with the other two.

Event boundary are important to determine the temporal relationships between two events. Automatic event detector

Table 4. Detection performance under varying mis-detection error

	interval level			frame level		
Noise intensity	30%	40%	50%	30%	40%	50%
CHSMM	0.90	0.73	0.59	0.82	0.70	0.57
DML-HMM	0.88	0.72	0.52	0.80	0.66	0.54
Our model	**0.90**	**0.75**	**0.66**	**0.85**	**0.73**	**0.62**

often makes mistakes in determining the start and end time of the event, as well as the event duration. In this experiment, we investigate the performance of our model under a varying event time measurement errors. We corrupted the testing data by perturbing the event start and end time by a noise with the noise intensity varying from $\pm 30\%$ to $\pm 50\%$ of the event duration. Table 5 shows the performance of the three models under different event time errors. It shows again that our model is more robust to the time measurement errors than CHSMM and DML-HMM, due to its ability in modeling the event duration and unsynchronized time lags between two temporal intervals.

Table 5. Detection performance under varying time error

	interval level			frame level		
Noise intensity	30%	40%	50%	30%	40%	50%
CHSMM	0.90	0.78	0.69	0.84	0.72	0.61
DML-HMM	0.91	0.77	0.62	0.82	0.68	0.58
Our model	**0.92**	**0.82**	**0.73**	**0.88**	**0.76**	**0.69**

7. Conclusions

We have proposed a temporal interval constrained DBN model for event detection in complex scenes. Allen's interval temporal relationships are successfully captured in our model to compensate for the poor image meansurements of the low-level visual detectors. An advanced structure learning algorithm has been presented to discover meaningful and salient dependencies in order to construct a computationally tractable network. Our model suits for the scenarios with multiple events occurring sequentially or in parallel, especially with overlapping event intervals, which forms complex relationships. Currently, our model only focuses on the atomic event detection. In the following work, we are interested in simultaneously detecting both the atomic event and the high-level complex activity in an unified model.

8. Acknowledgements

This work was partly supported by DARPA grant HR0011-08-C-0135-S8, 973 Program (2010CB327905), DARPA grant HR0011-10-C-0112, and National Natural Science Foundation of China (61202325).

References

[1] M. Albanese, R. Chellappa, V. Moscato, A. Picariello, V. S. Subrahmanian, and P. Turaga. A constained probabilistic Petri activity detection in video. *IEEE Trans. on Multimedia*, 10(8):1429–1443, 2008.

[2] J. F. Allen. Maintaining knowledge about temporal intervals. *Communication of the ACM*, 26(11):832–843, 1983.

[3] X. Boyen and D. Koller. Approximate learning of dynamic models. In *NIPS*, pages 396–402, 1999.

[4] W. Brendel, A. Fern, and S. Todorovic. Probabilistic event logic for interval-based event recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3329–3336. IEEE Computer Society, 2011.

[5] D. G. D. Heckerman and D. M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

[6] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching Hidden Semi-Markov Models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[7] S. Hongeng and R. Nevatia. Large-scale event detection using semi-hidden markov models. In *IEEE International Conference on Computer Vision*, 2003.

[8] T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behaviour in video. In *ICCV*, 2009.

[9] D. Kuettel, M. Breitenstein, L. V. Gool, and V. Ferrari. Whats going on? discovering spatio-temporal dependencies in dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[10] V. I. Morariu and L. S. Davis. Multi-agent event recognition in structured scenarios. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3289–3296. IEEE Computer Society, 2011.

[11] P. Natarajan and R. Nevatia. Coupled hidden semi markov models for activity recognition. In *IEEE Workshop on Motion and Video Computing*, 2007.

[12] N. M. Oliver, B. Rosario, and A. P. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.

[13] C. Pinhanez and A. Bobick. Human action detection using pnf propagation of temporal constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

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

[15] Y. Shi, A. F. Bobick, and I. A. Essa. Learning temporal sequence model from partially labeled data. In *CVPR (2)*, pages 1631–1638. IEEE Computer Society, 2006.

[16] P. Turaga, R. Chellappa, V. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Trans. on Circuits and Systems for Video Technology*, 18(11):1473–1488, 2008.

[17] J. Varadarajan, R. Emonet, and J. Odobez. Bridging the past,present and future: Modeling scene activities from event relationships and global rules. In *CVPR*, 2012.

[18] T. Xiang and S. Gong. Beyond tracking: Modeling activity and understanding behaviour. *International Journal of Computer Vision*, 67(1):21–51, 2006.