Modeling Temporal Interactions with Interval Temporal Bayesian Networks for Complex Activity Recognition

Yongmian Zhang, *Member*, *IEEE*, Yifan Zhang, *Member*, *IEEE*, Eran Swears, *Member*, *IEEE*, Natalia Larios, *Member*, *IEEE*, Ziheng Wang, *Student Member*, *IEEE*, and Qiang Ji, *Senior Member*, *IEEE*

Abstract—Complex activities typically consist of multiple primitive events happening in parallel or sequentially over a period of time. Understanding such activities requires recognizing not only each individual event but, more importantly, capturing their spatiotemporal dependencies over different time intervals. Most of the current graphical model-based approaches have several limitations. First, time-sliced graphical models such as hidden Markov models (HMMs) and dynamic Bayesian networks are typically based on points of time and they hence can only capture three temporal relations: precedes, follows, and equals. Second, HMMs are probabilistic finite-state machines that grow exponentially as the number of parallel events increases. Third, other approaches such as syntactic and description-based methods, while rich in modeling temporal relationships, do not have the expressive power to capture uncertainties. To address these issues, we introduce the interval temporal Bayesian network (ITBN), a novel graphical model that combines the Bayesian Network with the interval algebra to explicitly model the temporal dependencies over time intervals. Advanced machine learning methods are introduced to learn the ITBN model structure and parameters. Experimental results show that by reasoning with spatiotemporal dependencies, the proposed model leads to a significantly improved performance when modeling and recognizing complex activities involving both parallel and sequential events.

Index Terms—Activity recognition, temporal reasoning, Bayesian networks, interval temporal Bayesian networks

1 Introduction

Modeling and recognizing activities have undergone a rapid growth, starting from simple activities involving only a single entity to complex activities that involve multiple entities interacting with each other. A complex activity typically consists of multiple primitive events happening in parallel or sequentially over a period of time. Understanding such complex activities requires recognizing not only each individual event but also, more importantly, capturing their temporal dependencies. This is in particular the case when the detection of individual events is poor due to either poor tracking results, occlusion, background clutter, and so on.

Complex activity modeling and recognition is naturally solved by building a structure that is able to semantically capture the spatiotemporal relationships among events. Among various visual recognition methodologies, such as graphical, syntactic, and description-based approaches [1], time-sliced graphical models, i.e., hidden Markov models (HMMs) and dynamic Bayesian networks (DBNs), have become the most popular tool for modeling and understanding visual activities. Syntactic and description-based approaches have also gained attention in recent years for solving visual activity problems. However, these approaches face one or more of the following issues when modeling and understanding complex visual activities that involve interactions among different entities over durations of time:

1. Current graphical models are mostly time sliced (based on time points) and typically model events as occurring instantaneously, which is unrealistic for many real-world applications. Moreover, most models only offer three time-point relations (precedes, follows, equals); as such, they are not expressive enough to capture a larger number of temporal relationships between the events that happen over the duration of an activity.
2. The time-sliced graphical models used for activity modeling are probabilistic finite-state machines whose state-space grows exponentially in size with the number of parallel events [2], which quickly becomes intractable in both space and time for real-world complex activities.

- *Y. Zhang is with the IT Research Division, Konica Minolta Laboratory U.S.A. Inc., 2855 Campus Dr., San Mateo, CA 94403. E-mail: yongmianzhang@gmail.com.*
- *Y. Zhang is with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, and the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. E-mail: yfzhang@nlpr.ia.ac.cn.*
- *E. Swears, N. Larios, Z. Wang, and Q. Ji are with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180. E-mail: eran.swears@kitware.com, nlarios@microsoft.com, wangz10@rpi.edu, qji@ecse.rpi.edu.*

Manuscript received 29 Mar. 2012; revised 24 Sept. 2012; accepted 8 Jan. 2013; published online 24 Jan. 2013.
Recommended for acceptance by V. Pavlovic.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2012-03-0231.
Digital Object Identifier no. 10.1109/TPAMI.2013.33.

3. Syntactic and description-based models lack the expressive power to capture and propagate the uncertainties associated with event detection and with their temporal dependencies in a principled manner.

To address these issues, we propose a unified probabilistic framework that combines the probabilistic semantics of the Bayesian networks (BNs) [3] with the temporal semantics of interval algebra [4] (IA). Termed interval temporal Bayesian network (ITBN), this framework employs the BNs' probabilistic basis and the IA's temporal relational basis in a unified model that allows representing not only the spatial dependencies among events but also a larger variety of time-constrained relations, while remaining fully probabilistic and expressive of uncertainty. In contrast with time-sliced graphical models, the ITBN model is time interval-based (instead of time point). ITBN is effective in uncertainty representation and propagation and in representing a full range of temporal relations between events, characterizing both parallel and sequential interactions over multiple durations of time.

The remainder of this paper is organized as follows: Section 2 presents an overview of the existing work in activity recognition and discusses their similarities and differences to our work. An introduction to Allen's IA and the temporal nature of events are provided in Section 3. Section 3 also introduces the details of our ITBN model. ITBN parameter and structure learning is covered in Section 4. This is then followed by discussing activity modeling and understanding with ITBNs in Section 5 and the experimental results and analysis are presented in Sections 6 and 7. The final section provides a summary of this work and the concluding remarks.

2 Related Work

Modeling visual activities had an early start with human action recognition. This initial work was concentrated on understanding the movements of the human body or individual body parts of a single person in a video sequence. A thorough survey on action recognition can be found in [1]. Significant progress has been made in the last decade in activity modeling and understanding, most notably when modeling and understanding complex scenes involving multiple interacting objects. This progress has been enabled in part by employing time-sliced graphical models such as HMMs and its variants. These models have become central to modeling such activities. Most of these models, however, focus on modeling concurrent interactions among entities. For example, Oliver et al. [5] exploited coupled hidden Markov models (CHMMs) [6] to model basic human interactions such as one person following another, altering their path to meet another, and so forth. To capture interactions, CHMMs have multiple state variables that are temporally coupled through the conditional probabilities of one chain given the other chain. Similarly, Park and Aggarwal [7] proposed recognizing concurrent human interactions using a hierarchical framework consisting of coupled DBNs/BNs. Hamid [8] also used DBNs for recognizing interacting multi-agent activities, where the interactions between objects are characterized by low-level spatiotemporal features such as the relative distances between objects and agents as well as their velocities. Such characterization of the interactions at the observation level, however, requires reliable detection and tracking of each entity.

To handle sequential interactions, Xiang and Gong [9], [10] presented a dynamically multilinked hidden Markov model (DML-HMM) for modeling the temporal and causal correlations among events of an activity in an outdoor scene. The topology of the DML-HMM is built through the discovery of salient dynamic interlinks among multiple temporal processes corresponding to multiple event classes. Duong et al. [11] proposed a switching hidden semi-Markov model (S-HSMM) for recognizing a sequence of events. The bottom layer of the S-HSMM represents atomic events and the top layer represents a sequence of high-level activities, where each high-level activity is comprised of a sequence of atomic events. Like the DML-HMM model, the relationships among events in a S-HSMM model are limited to simple sequential relationships such as before or after. Propagation nets (P-Net) by Shi et al. [12] are a DBN extension that models duration for complex activity representation, including concurrent events. P-Nets require manually specified links between states.

Various hybrid temporal models have also been proposed to capture the temporal relationships among events. These models include the causal temporal constraint networks (CTCNs) [13], the temporal logic [14], [15], [16], [17], [18], the modifiable temporal Bayesian network (MTBNs) [19], the probabilistic temporal networks (PTNs) [20] (also named temporal BNs [21]), and the multi-agent networks [22]. These models share the basic similarity of employing some form of temporal information. Nevertheless, the specific semantics and structure each model network are quite different, along with their modeling power and associated algorithms. CTCNs and temporal logic use qualitative logic for inference, but lack the probabilistic semantics that are needed to represent uncertainty in temporal dependencies. MTBNs are primarily an extension to time-sliced BN defined over a range of time points.

Among these temporal models, PTNs and the multi-agent networks are the most similar to ITBN. PTNs are directed acyclic graphs (DAG) whose nodes represent temporal aggregates (TAs) that contain a set of RV-interval pairs. The edges represent temporal causal relationships (TCR) between aggregates. Each TCR is a shorthand for a set of induced random variables that capture causal temporal relationships between two TAs. Of the 13 interval relations shown in Fig. 2, PTNs can only represent 7. This limits their application mostly to periodic and recurrent activities. In addition, these relations are treated as known observations. And, the PTN implementation described in [21], [20] lacks experimental validation and a mechanism for automated structure and parameter learning from training data. Finally, instead of using a probabilistic inference, PTN inference is carried out using an alternative formulation based on solving a linear constraint system [23]. Intille and Bobick [22] proposed a multi-agent network for modeling multiple agents interacting in parallel event

streams consisting of collaborative actions (i.e., football plays). Like the proposed ITBN model, the network employs nodes in a BN graph to represent temporal relationships. It, however, differs from ITBN in several aspects. First, the ITBN can represent Allen's 13 temporal relationships, while multi-agent networks can only capture two temporal relationships (before and overlap). Hence, the ITBN is able to discriminate between more complex activities. Second, the network structures and parameters are rigorously learned from training data for the ITBN, while they are manually specified for the multi-agent network. Third, it requires the use of non-Bayesian modularity to facilitate the building and the inference on the multi-agent networks, which contrasts starkly with the principled and rigorous learning and inference methods for ITBN. Finally, besides recognizing activities, the ITBN model can also reason about the relationships between two events given the state of other events and/or their relationships.

First-order logic (FOL) allows us to compactly represent a wide variety of knowledge and is combined with probability to form the *first-order probabilistic languages* [24], [25], [26]. Among the relevant models in this category are relational dynamic Bayesian networks (RDBNs) [27], Bayesian logic (BLOG) [28], and Markov logic networks (MLN) [29], [25]. RDBNs are combination of DBNs with a subset of FOL (i.e., ground predicates) that can handle time-changing phenomena and uncertainty in a relational domain. However, RDBNs remain a time-sliced model that cannot effectively handle relations occurring over time intervals. BLOG seeks to define a formal language to express probability models that explicitly represent an unknown number of objects and their relations. This property of BLOG is relevant in complex activity modeling because some activity categories are composed of a varying or unknown number of interacting objects (e.g., a street rally, vehicle tracking). BLOG, however, faces challenges on inference decidability, evaluating well-formed models, and on structure learning. Moreover, to the best knowledge of the authors, there is no experimental evaluation using BLOG for activity recognition.

An MLN model is a first-order logic knowledge base with a weight attached to each formula and with no restriction other than the finiteness of the domain. Inference is often performed by a Markov chain Monte Carlo method. Morariu and Davis [25] propose an MLN-based approach for complex multi-agent event recognition that employs interval-based [4] knowledge such as rules, event descriptions, and physical constraints of the events being modeled. The MLN knowledge base and formula weights are intuitively defined by experts, with a high cost for breaking physical laws and a small penalty for breaking a rule of thumb. Clearly, this method is mainly knowledge driven. The related rules and relations are manually encoded into the first-order logic formulas. Also, as an undirected graphical model, the structure of MLN is typically manually specified, which is quite different from BN-based methods. Generally, this method focuses on recognizing simple actions typically performed by one person. Direct extension of this method to recognize activities with higher complexity could risk complex modeling and, hence, computationally expensive inference.

Event logic (EL) [30] is a specialized representation of temporal knowledge grounded on interval-based events [4]. Probabilistic event logic (PEL), proposed by Brendel et al. [26], is a probabilistic treatment of EL based on confidence-weighted formulas, similarly as MLN is to first-order logic. It has been applied to improve detection and label assignment of the primitive events. However, the temporal relationships between the primitive events must be known in advance to manually encode them into the EL formulas. In this approach, costly interval enumeration is avoided by using spanning intervals. A set-based representation of groups of intervals is proposed to greatly reduce the number of intervals to consider during inference.

In summary, compared with the FOL methods, ITBN is more data driven, while MLN and PEL approaches are more knowledge driven. ITBN is therefore expected to outperform MLN and PEL when the domain knowledge of the primitive events and their spatiotemporal relationships are unknown in advance, while it would underperform if the training data are not sufficient. In addition, ITBN is more suitable for modeling complex activities composed of multiple interactive primitive events. MLN and PEL approaches, on the other hand, mainly focus on improving the event hypotheses generated by the noisy observations. They could be extended to recognizing complex activities, but that would risk expensive computational cost.

Besides graphical models, several authors have attempted to use the probabilistic Petri nets (PPNs), suffix trees, finite-state automata (FSA), context-free grammar (CFG), and stochastic context-free grammar for modeling activities involving multi-object interactions. For example, PPNs were used by Albanese et al. [31] to model an activity composed of sequential and concurrent atomic actions. Hamid et al. [32] considered a temporally extended activity as a sequence of events that follow some inherent partial ordering. Using these constraints, they consider an activity model as a set of subsequences which encode partial ordering constraints of varying lengths. These subsequences are efficiently represented using suffix trees. CFG approaches present a sound theoretical basis for modeling structured processes. Ryoo and Aggarwal [33], as well as Ivanov and Bobick [34], used CFG to model and recognize composite human activities and multiperson interactions. They followed a hierarchical approach, where the lower levels are composed of HMMs and the higher level model interactions with CFGs. The overall recognition process is performed by parsing with stochastic production rules. Hongeng et al. [17] and Hakeem and Shah [35] used FSA states to represent the events of an activity. Most recently, Gupta et al. [36] proposed a storyline model with an AND-OR graph to determine the story of a video by exploiting spatiotemporal relationships between actions, where the temporal orderings of actions are also used as a hard constraint to define causal dependencies among actions.

Besides the models mentioned above, topic model and its variants have also been successfully applied to activity recognition lately. Kuettel et al. [37] proposed a DDP-HMM model to discover the activities in the traffic scenes. Instead of tracking each primitive event, they proposed to directly model each activity as a bag of flow words that occur along

certain trajectories. In this way, each event corresponds to a topic which is a specific spatial flow pattern. And the events were automatically learned with a hierarchical Dirichlet process from the computed flow features. Spatiotemporal relations among different events are captured with an HMM. The method is robust and can recognize different traffic behaviors from real traffic data. Hospedals et al. [38] introduced a weakly supervised joint topic model to recognize rare and subtle behaviors from the traffic scenes. A similar method is used to extract visual words based on the flow features. To address the inadequate training data problem for the rare events, the learned topics are shared among different classes of activities.

The benefits of topic models are that they do not need perform explicit tracking and event detection. They can therefore maintain robustness to the widespread challenges found in activity modeling like tracking error and event duration errors. Such flow-based methods, however, are susceptible to background motion and camera ego motion. In addition, computing the dense optical flows and calculating the visual words is expensive and time consuming. Compared with the proposed ITBN model, the topic model-based methods cannot effectively model and recognize activities with strong and diverse temporal relationships among the underlying temporal entities.

In summary, the majority of work in modeling complex activities employs HMMs, DBNs, or their variants. Since they are all time-sliced graphical models defined over points of time, the temporal relationships among events are limited to point-based relationships, including before, during, and after. These models are therefore unable to model complex activities that involve duration, parallel, and multithreading events. On the other hand, current interval-based approaches employing probabilistic treatment of logic have focused on solving primitive event tasks, and they 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. These approaches could encounter intractable and complex inference when modeling high-level complex activities. Finite-state automatons have difficulty scaling up to complex activities involving parallel events and simultaneous occurrences. Other models such as PPNs and the suffix trees are intuitive tools for modeling complex activities. However, these description-based approaches lack a rigorous mechanism for automatically learning the structure and parameters from training data. Additionally, they lack the expressive power to model uncertainty and hence are often designed to be deterministic. Both stochastic grammar parsing and FSA have difficulty handling complex activities, object interactions, and missing observations. The topic models, while powerful in recognizing some complex activities, cannot effectively handle activities with strong and diverse temporal dependencies. The proposed approach attempts to explicitly address these limitations.

3 Interval Temporal Bayesian Networks

3.1 Interval Algebra

An event is defined as the state change of one or more entities over a period of time, while an activity is a collection of temporally and coherently related events. Events occur over intervals of time and are correlated by their temporal relationships, and these temporal relationships over time, taken together, constitute a regular rhythmic pattern, i.e., the activity. Consider a cargo loading scenario [9], as shown in Fig. 1. Cargo loading is an activity that consists of seven events:

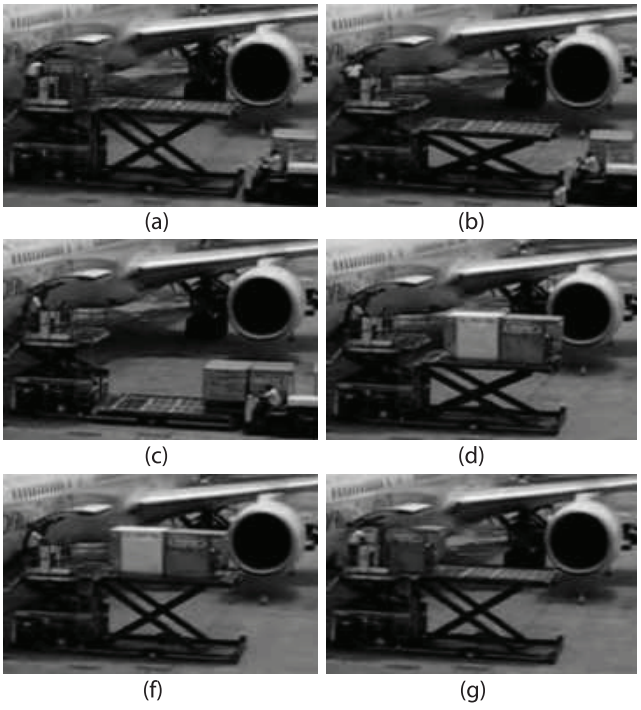


Fig. 1. An activity of loading cargo into an airplane, where only six atomic events are shown in this figure: (a) cargo truck approaching, (b) cargo lift lowering down to the loading position, (c) unloading container to the cargo lift, (d) cargo lift lifting up to the unloading position, (e) cargo truck leaving (not shown in this figure), (f) unloading container from the cargo-lift, and (g) conveying the container into the airplane.

1. cargo truck approaching,
2. lowering the cargo lift to the loading position,
3. unloading the container to the cargo lift,
4. lifting up the cargo lift to the unloading position,
5. cargo truck leaving,
6. unloading the container from the cargo lift, and
7. conveying the container into the airplane.

Each of the events consists of one or more entities and happens over a period of time. The events happen in either a sequential or parallel manner to complete the activity. For example, lowering the cargo lift from the unloading position to the loading position usually takes about 30 seconds, and it often happens during the cargo truck approaches, but it must occur before the container is unloaded. It can be seen that the notion of temporally dependent events over multiple time intervals plays an essential role in capturing the essence of complex activities.

According to Allen's axiomatization of time periods [39], there are 13 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. 2, 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

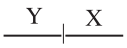
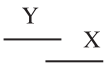
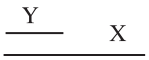
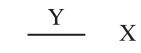
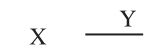
Relation	Symbol	Inverse	Pictoral Meaning
Y before X	b	bi	
Y meets X	m	mi	
Y overlaps X	o	oi	
Y starts X	s	si	
Y during X	d	di	
Y finishes X	f	fi	
Y equal X	eq	eq	

Fig. 2. Allen's 13 atomic interval temporal relations to represent the temporal relations between two events X and Y.

atomic relations, e.g., $Y\{b,m\}X$ representing $(Y\ before\ X)$ or $(Y\ meets\ X)$. An interval algebra network (IAN) [39] can be used to represent the temporal relationships among a set of events in an activity, 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 example that models the interval temporal relationships among three events in the cargo loading activity.

The IAN is effective in capturing temporal relationships occurring over multiple time intervals. However, the occurrence of an event and its temporal relationships with other events are often uncertain. Thus, an activity model must be able to handle uncertainties. Unfortunately, despite its capability to effectively capture a range of temporal relationships, IAN does not support reasoning and inference under uncertainty, which limits its capability for activity modeling.

3.2 Interval Temporal Bayesian Network

BNs [3] have been increasingly used in different applications for modeling the probabilistic relationships among random variables. BNs capture the conditional dependencies among random variables via a DAG. They provide a probabilistic method for representing and propagating uncertainties and for reasoning under uncertainty. As a temporal extension to BNs, the DBNs are widely used to model dynamic processes and to perform reasoning under uncertainty over time. DBNs generalize popular dynamic models such as HMM and Kalman filtering which have been successfully used in computer vision. However, as a time-sliced model with first-order Markovian assumption, a DBN model lacks the capability of representing different interval temporal relationships between the events over different time durations. In fact, the temporal relationships captured by a DBN are limited to such point-based relationships as before, during, and after. To address the shortcomings with the BNs and with the IANs and to continue exploiting their respective strengths, we propose a unified framework, the ITBN, based on combining the IANs with the BNs. By unifying the modeling characteristics of

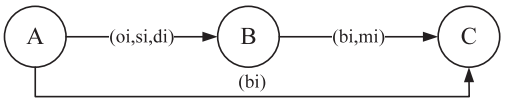


Fig. 3. An example of Allen's IAN modeling the interval temporal relationships among three events of the cargo loading activity. A = cargo truck approaching, B = cargo lift lowering to the loading position, C = unloading container to the cargo lift.

BNs with IANs, an ITBN can take advantage of BN's power in statistical relationship representation and reasoning as well as IAN's capability of representing different types of interval temporal relationships between events. This unified model is more expressive in modeling complex activities than its counterparts. In the paragraphs below, we will first introduce some definitions, based on which we will then formally introduce the ITBN approach.

Definition 1 (Temporal Entity). *A temporal entity is characterized by a pair $\langle \Sigma, \Omega \rangle$ in which Σ is a set of all possible outcomes for the temporal entity and Ω is a period of time spanned by the temporal entity and $\Omega = \{[a,b] \in \mathbb{R}^2 : a < b\}$, where a and b denote the start time and the end time, respectively.*

When the temporal entities represent the events of an activity, Σ is the state of event occurrence and Ω shall be the time interval spanned by the event.

Definition 2 (Temporal Reference). *If a temporal entity X is used as a time interval reference for determining temporal relations to another temporal entity Y, then X is a temporal reference of Y.*

Definition 3 (Temporal Dependency (TD)). *A TD denoted as $I_{X,Y}$ describes temporal relationships between two temporal entities $X = \langle \Sigma_X, \Omega_X \rangle$ and $Y = \langle \Sigma_Y, \Omega_Y \rangle$. As shown in Fig. 4a, $I_{X,Y}$ is graphically represented as a directed link leading from the node X to the node Y labeled with $I_{X,Y} \in \mathbf{R} = \{b, bi, m, mi, o, oi, s, si, d, di, f, fi, eq\}$, where X is the temporal reference of Y. The strength of the TD can be quantified by a conditional probability as follows:*

$$P(I_{X,Y} = i | X = x, Y = y), \tag{1}$$

where $x \in \Sigma_X$ and $y \in \Sigma_Y$ are the states of the temporal entities and $i \in \mathbf{R}$ denotes an interval temporal relation.

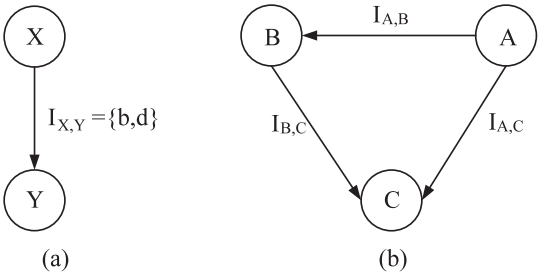


Fig. 4. (a) A graphical representation of TD between temporal entities X and Y is captured by a directed link from the temporal reference X to the temporal entity Y and the link is labeled with the interval temporal relation $I_{X,Y} = \{b, d\}$. (b) An example of the ITBN having three temporal entities with interval temporal relationships $I_{A,B}$, $I_{A,C}$, and $I_{B,C}$, respectively.

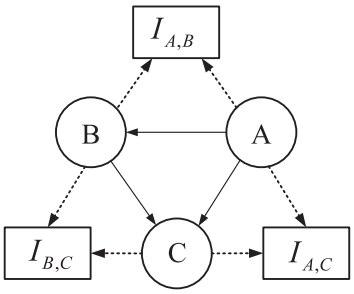


Fig. 5. The BN equivalent graphical representation of the ITBN model shown in Fig. 4b, where a circled node denotes a temporal entity and a squared node represents its temporal relationship with respect to another entity. The solid links represent spatial dependencies among temporal entities, while the dotted links capture the temporal dependencies among the temporal entities.

Here, we only consider pairwise temporal dependencies. Given these definitions, we can formally introduce the ITBN as follows:

Definition 4 (Interval Temporal Bayesian Network). *An ITBN is a* DAG $G(V, E)$, *where* V *is a set of nodes representing temporal entities and* E *is a set of links representing both the spatial and temporal dependencies among the temporal entities in* V.

A link in an ITBN is a carrier of the interval temporal relationship, and the link direction leading from X to Y indicates Y is temporally dependent on X, and X is the temporal reference of Y. Once the temporal reference is established, the direction of the arc cannot be changed. It can only point from the temporal reference to the other temporal entity, therefore avoiding the temporal relationship ambiguity. The strength of the TD is quantified by the forward conditional probabilities as given in (1). An example of a simple ITBN can be seen in Fig. 4b.

We propose to implement ITBNs with a corresponding BN to exploit the well-developed BN mathematical machinery. Fig. 5 shows the corresponding BN graphical representation for the ITBN shown in Fig. 4b, where another set of nodes (the square nodes) is introduced to represent the temporal relationships. Specifically, an ITBN implemented as a BN includes two types of nodes: temporal entity nodes (circular) and temporal relationship nodes (square). There are also two types of links, spatial links (solid lines) and temporal links (dotted lines). The spatial links connect among the temporal entity nodes and they capture the spatial dependencies among the temporal entities. The temporal links connect the temporal relationship nodes with the corresponding temporal entity nodes and they characterize the temporal relationships between the two connected temporal entities. Given this representation and following the local Markov properties of a BN, the joint probability of the nodes and the links in an ITBN can be factorized as the product of the conditional probabilities of the temporal entity nodes and the conditional probabilities of the temporal relation nodes, i.e.,

$$P(\mathcal{Y}, \mathcal{I}) = \prod_{j}^{n} P(Y_j|\pi(Y_j)) \prod_{k}^{K} P(I_k|\pi(I_k)), \tag{2}$$

where $\mathcal{Y} = \{Y_j\}_{j=1}^{n}$ and $\mathcal{I} = \{I_k\}_{k=1}^{K}$ represent all temporal entity nodes and all temporal relation nodes, respectively, in an ITBN. $\pi(Y_j)$ is the set of parental nodes of Y_j; I_k represents the kth interval temporal relation node and $\pi(I_k)$ are the two temporal entity nodes that produce I_k. The ITBN's parameter vector, $\Theta = (P(Y_j|\pi(Y_j), P(I_k|\pi(I_k))))$, includes the conditional probabilities of the temporal entity nodes and the conditional probabilities of the temporal relation nodes.

4 ITBN Learning

To use the ITBN model for an application, we need to first construct the model from the available training data. The following section discusses how to learn the parameters, the interval temporal relationships, and the network structure of an ITBN from training data.

4.1 Parameter Learning

Assume a training data set $D = \{D_1, \ldots, D_m, \ldots, D_M\}$, where all n nodes are fully observed on each exemplar, D_m. Also, it is assumed that the interval temporal relationships I_k between a node and its reference node have been established and properly labeled. The goal of parameter learning is to find the maximum likelihood estimate (MLE) of the parameters $\Theta = \{\Theta_1, \ldots, \Theta_n\}$ for the given training data D. The ITBN parameters include the conditional probability for each temporal entity node, i.e., $\Theta_j^e = P(Y_j \mid \pi(Y_j))$, where $j = 1, 2, \ldots, n$, and the conditional probability for each temporal relation node, i.e, $\Theta_k^r = P(I_k|\pi(I_k))$, where $k = 1, 2, \ldots, K$. Assuming the samples are independent and identically distributed, the joint likelihood of the parameters with respect to the training data can be defined as

$$\begin{aligned} L(\Theta : D) &= \prod_{m}^{M} P(\mathcal{Y}[m], \mathcal{I}[m] : \Theta) \\ &= \prod_{m}^{M} \prod_{j}^{n} P\big(Y_j[m]|\pi(Y_j)[m] : \Theta_j^e\big) \\ &\quad \prod_{m}^{M} \prod_{k}^{K} P\big(I_k[m]|\pi(I_k)[m] : \Theta_k^r\big). \end{aligned} \tag{3}$$

Let $L_j(\Theta_j^e : D)$ be the joint likelihood of the conditional probability of temporal entity node j for all samples and $L_k(\Theta_k^r : D)$ be the joint likelihood of the conditional probabilities of temporal relation node k for all samples. Equation (3) can be further reduced to

$$L(\Theta : D) = \prod_{j}^{n} L_j\big(\Theta_j^e : D\big) \prod_{k}^{K} L_k\big(\Theta_k^r : D\big). \tag{4}$$

There are two independent estimation problems: estimation of conditional probabilities Θ_j^e and estimation of conditional probabilities Θ_k^r. It is assumed that the temporal entities are binary states $\{1, 0\}$ and the parameter $\Theta_j^e = P(Y_j|\pi(Y_j))$ has a multinomial distribution. The estimation of Θ_j^e then becomes

TABLE 1
Interval Relation Defined by Event Starting and Finishing Time

No.	r	$d(a_Y, a_X)$	$d(b_Y, b_X)$	$d(a_Y, b_X)$	$d(b_Y, a_X)$
1	b	< 0	< 0	< 0	< 0
2	bi	> 0	> 0	> 0	> 0
3	d	> 0	< 0	< 0	> 0
4	di	< 0	> 0	< 0	> 0
5	o	< 0	< 0	< 0	> 0
6	oi	> 0	> 0	< 0	> 0
7	m	< 0	< 0	< 0	$= 0$
8	mi	> 0	> 0	$= 0$	> 0
9	s	$= 0$	< 0	< 0	> 0
10	si	$= 0$	> 0	< 0	> 0
11	f	> 0	$= 0$	< 0	> 0
12	fi	< 0	$= 0$	< 0	> 0
13	eq	$= 0$	$= 0$	-	-

Note: X is the temporal reference of Y.

$$\begin{aligned} L_j(\Theta_j^e : D) &= \prod_m^M P\big(Y_j[m] | \pi(Y_j)[m] : \Theta_j^e\big) \\ &= \prod_{\pi(Y_j)} \prod_{Y_j} P\big(Y_j | \pi(Y_j) : \Theta_j^e\big)^{N(Y_j, \pi(Y_j))} \\ &= \prod_{k \in 2^{|\pi(Y_j)|}} \prod_{l \in \{1,0\}} \theta_{Y_j = l | \pi(Y_j) = k}^{N(Y_j = l, \pi(Y_j) = k)}, \end{aligned} \tag{5}$$

where Θ_j^e is a vector that contains parameter θ_{Y_j} for each value of $\pi(Y_j)$ and Y_j. $N(Y_j = l, \pi(Y_j) = k)$ is the number of times the event $(Y_j = l, \pi(Y_j) = k)$ occurred in the training dataset. Equation (5) decomposes the likelihood function into an independent multinomial problem. By taking its log, adding a Lagrange multiplier to ensure $\sum_l \theta_{Y_j = l | \pi(Y_j) = k} = 1$, and setting the partial derivatives to zero, we can readily obtain the MLE:

$$\hat{\theta}_{Y_j = l | \pi(Y_j) = k} = \frac{N(Y_j = l, \pi(Y_j) = k)}{N(\pi(Y_j) = k)}, \tag{6}$$

where $N(\pi(Y_j) = k)$ is the number of times the event $\pi(Y_j) = k$ occurred in the training dataset. Analogously, applying the MLE principle to the likelihood function $L_k(\Theta_k^r : D)$, we get an MLE estimate of the conditional probability $\theta_{I_k} = P(I_k = i | \pi(I_k))$ as

$$\hat{\theta}_{I_k = i} = \frac{N(I_k = i)}{\sum_{i \in R} N(I_k = i)}, \tag{7}$$

where $N(I_k = i)$ is the count of the ith temporal relationships in the training data for the kth temporal relationship node when both parents of I_k are present. Note $P(I_k | \pi(I_k))$ is set to be uniform if either one or both of the parents of I_k are not present.

4.2 Structure Learning

In practice, interval temporal relationships among events for an activity are not known in advance, so they need to be statistically learned from the training data before the network parameters and structure can be learned. Thus, learning the network structure consists of two steps: 1) learning (labeling) the interval temporal relationships and then 2) learning the network structure.

Table 1 defines the interval temporal relation r using the temporal distance $d(\Omega_Y, \Omega_X)$ between two entities X and Y, where X is the temporal reference of Y, Ω is the time duration of the event as defined in Definition 1. We define the temporal distance as the distance of two time intervals $[a_X, b_X]$ and $[a_Y, b_Y]$, where X and Y are two events, as follows:

$$d(\Omega_Y, \Omega_X) = \{a_Y - a_X, b_Y - b_X, a_Y - b_X, b_Y - a_X\}. \tag{8}$$

Note that because of the detection errors with event times, it may be difficult to know the precise starting and ending times of an event. Therefore, measurement error shall be considered in determining the temporal distances. An estimate of the measurement error can be obtained by comparing the detected event endpoint times with those of the available ground truth data. Using the estimated detection error, we can then set a threshold to classify if two times are before, after, or equal to each other, as required in Table 1.

Given a training dataset with M independent examples and n entities, where the training data includes the time period during which an event occurs. The interval temporal relationships between all pairwise entities can be learned through the procedure given in Table 2. The procedure computes an $n \times n$ array denoted as T, where n is the number of events. Each cell $T_{i,j}$ is the union of all the temporal relationships I_{ij} found between the two events i and j in the data. Table 3 gives an example of the interval temporal relationships between pairwise events for the cargo loading activity.

ITBNs shall not only be DAG consistent but also temporally consistent. The mechanism behind the temporal consistency test is as follows: Choose any three vertices X, Y, and Z in the ITBN that completes the triangle $\triangle XYZ$, as shown in Fig. 6. Then, $I_{Z,X}$ is constrained by $I_{Y,X}$ and $I_{Z,Y}$. For example, assuming $I_{Y,X} = \{m\}$ and $I_{Z,Y} = \{di\}$, we can deduce $I_{Z,X} = \{b\}$. Such a transitivity property can be expressed as

$$\begin{aligned} I_{Z,X} &= I_{Y,X} \circ I_{Z,Y} \\ &= \{A(i_1, i_2), \forall i_1 \in I_{Y,X}, \forall i_2 \in I_{Z,Y}\}, \end{aligned} \tag{9}$$

TABLE 2
Interval Temporal Relation Algorithm for All Entity Pairs

For $i = 1, \cdots, n$, where n is the number of event nodes
  For $j = 1, \cdots, n$
    $I_{i,j} = \emptyset$, initialize to an empty set
    For $k = 1 \cdots M$, where M is the sample size
      $r =$ the entry of Table 1 based on $d_k(\Omega_i, \Omega_j)$
      If $r \notin I_{i,j}$
        $I_{i,j} = I_{i,j} \cup r$
Return I, where I is an $n \times n$ table of cells

TABLE 3
An Example of Interval Temporal Relations between Pairwise Events for Cargo Loading Activity

	A	B	C	D	E	F
A		o,s,d	b	b	b	b
B	oi,si,di		b,m	b	b	b
C	bi	bi,mi		m,b	m,b	b
D	bi	bi	mi,bi		s,o,d	b,m
E	bi	bi	mi,bi	si,oi,di		o,fi,di
F	bi	bi	bi	bi,mi	oi,f,d	

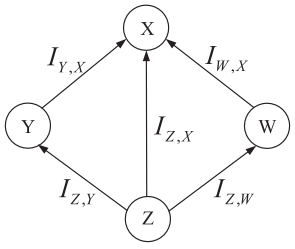


Fig. 6. $\triangle XYZ$ and $\triangle XWZ$ are the triangles of temporal dependencies having a common temporal link I_{ZX}, where I_{ZX} should be consistent with other temporal relationships

where $\circ$ denotes a composition operation and $A(i_1, i_2)$ is the entry of the transitivity lookup table shown in Table 4 row i_1 and column i_2, where i_1 represents the relation between X and Y, with Y being the temporal reference of X, and i_2 the relation between Y and Z, with Z being the temporal reference of Y.

In other words, $I_{Z,X}$ is the union of all $A(i_1, i_2)$. Table 4 shows the transitivity lookup table adapted directly from [39]. The interval temporal relations of a common temporal link (shared by multiple triangles, i.e., $I_{Z,X}$ in Fig. 6) can be deduced from

$$I_{Z,X} = (I_{Y,X} \circ I_{Z,Y}) \cap (I_{W,X} \circ I_{Z,W}). \quad (10)$$

If $I_{Z,X} = \emptyset$, then there shall not be a temporal link between X and Z. This structural property will be utilized for structure learning, as discussed in the next section, which ensures that all the triangles formed with temporal dependencies are temporally consistent. The interval temporal relations for the edge of a triangle in an ITBN shall be

$$I_{Z,X} = I^D_{Z,X} \cap I^L_{Z,X}, \quad (11)$$

where $I^D_{Z,X}$ is a set of temporal relations deduced by using (9) and (10), and $I^L_{Z,X}$ a set of relations learned from training data as described previously. Once the temporal relationships of a model are established, the network parameters and structure can be learned from the same set of training data.

Learning the ITBN structure means finding a network, G, that best matches the training data set D. We use Bayesian information criterion (BIC) [40], to evaluate each ITBN:

TABLE 5
ITBN Structure Learning Algorithm

1: Initialize structure to G_0 with domain knowledge or randomly
2: Learn temporal relations for all pairwise events using Table 2
3: Estimate parameters Θ_0 and score S_0 for G_0
4: For $n = 0, 1, \cdots$ until converge
5: Get G_{n+1} by adding, removing or reversing one link from G_n
6: If G_{n+1} satisfy DAG and TC
7: Estimate parameters Θ_{n+1} and score S_{n+1} for G_{n+1}
8: If $S_{n+1} > S_n$ Then $G^* = G_{n+1}$
9: Return G^*

$$\max_G S(G : D) = \max_\Theta \left(\log P(D|G, \Theta) - |\Theta| \frac{\log N}{2} \right), \quad (12)$$

where S denotes a BIC score, Θ the vector of the estimated parameters, $\log P(D|G, \Theta)$ the log-likelihood function, and $|\Theta|$ the number of free parameters. We utilize a local search procedure [41], [42] that changes one arc (insertion, deletion, and reversal) on each iteration. In addition, to ensure that the ITBN is DAG consistent, the temporal consistency constraint must also be satisfied, i.e, any newly formed triangle in the ITBN due to arc addition shall be temporally consistent, and the consistency can be tested with (9) and (10). Unnecessary relationships and temporally inconsistent links shall be removed.

On each iteration, the search process changes one link to produce a candidate ITBN G. Before evaluating G with S using (12), the newly formed triangles need to be tested for temporal consistency. If G does not pass the test, it is then rejected and the next model is evaluated. The ITBN structure learning algorithm is summarized in Table 5.

5 Visual Activity Modeling

In this section, we describe how the activity modeling problem fits naturally within the proposed ITBN framework. A complex activity typically involves multiple primitive events performing various interactions in the same scene; such activity can be described as a group of temporally and spatially correlated events with a hierarchical nature. These relationships can be effectively modeled with an ITBN.

TABLE 4
The Transitivity Table for Atomic Interval Temporal Relations (Adapted from [39])

i_2 / i_1	b	bi	d	di	o	oi	m	mi	s	si	f	fi	eq
b	b	no info.	b,o,m,d,s	b	b	b,o,m,d,s	b	b,o,m,d,s	b	b	b,o,m,d,s	b	b
bi	no info.	bi	bi,oi, mi,d,f	bi	bi,oi,mi, d,f	bi	bi,oi,mi, d,f	bi	bi,oi, mi,d,f	bi	bi	bi	bi
d	b	bi	d	no info.	b,o,m,d,s	bi,oi,mi,d,f	b	bi	d	bi,oi,mi,d,f	d	b,o,m,d,s	d
di	b,o,m, di,fi	bi,oi,di, mi,si	o,oi,d,s,f, di,si,fi,eq	di	o,di, fi	oi,di, si	o,di, fi	oi,di, si	di,fi,o	di	di,si,oi	di	di
o	b	bi,oi,di, mi,si	o,d,s	b,o,m,di,fi	b,o,m	o,oi,d,s,f, di,si,fi,eq	b	oi,di,si	o	di,fi,o	d,s,o	b,o,m	o
oi	b,o,m,di,fi	bi	oi,d,f	bi,oi,mi, di,si	o,oi,d,s,f, di,si,fi,eq	bi,oi,mi	o,di,fi	bi	oi,d,f	oi,bi,mi	oi	oi,di,si	oi
m	b	bi,oi,mi,di,si	o,d,s	b	b	o,d,s	b	f,fi,eq	m	m	d,s,o	b	m
mi	b,o,m, di,fi	bi	oi,d,f	bi	oi,d,f	bi	s,si,eq	bi	d,f,oi	bi	mi	mi	mi
s	b	bi	d	b,o,m,di,fi	b,o,m	oi,d,f	b	mi	s	s,si,eq	d	b,m,o	s
si	b,o,m,di,fi	bi	oi,d,f	di	o,di,fi	oi	o,di,fi	mi	s,si,eq	si	oi	di	si
f	b	bi	d	bi,oi,mi,di,si	o,d,s	b,oi,m	m	bi	d	bi,oi,mi	f	f,fi	f
fi	b	bi,oi,mi,di,si	o,d,s	di	o	oi,di,si	m	si,oi,di	o	di	f,fi,eq	fi	fi
eq	b	bi	d	di	o	oi	m	mi	s	si	f	fi	eq

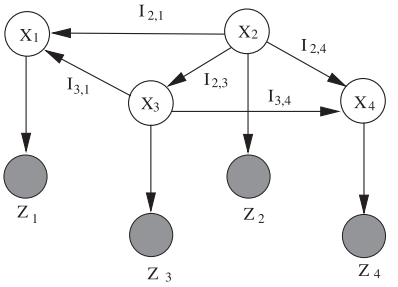


Fig. 7. An example ITBN activity model where we assume that the activity consists of four discrete events X_1, X_2, X_3, and X_4. The shaded nodes Z_1, Z_2, Z_3, and Z_4 are the observations for the events X_1, X_2, X_3, and X_4, respectively. $I_{i,j}$ is a set of interval relationships between X_i and X_j, where X_i is the temporal reference.

Fig. 7 shows a general two-layer ITBN model. The top layer encodes the primitive events and their spatial and temporal dependencies, where each node represents an event, and the event spatial and temporal relationships are captured by the links and their labels. The links are quantified by the conditional probabilities. The nodes in this layer are hidden and so must be inferred from the observations in the bottom layer. The bottom layer is comprised of a set of observation nodes Z_j that ingest event detections for their corresponding event node X_j in the top layer. The observations include information about the presence or absence of the event and its starting and ending times. When an event is detected, the time interval spanned by the event is also recorded to estimate the pairwise interval relations between events.

Having learned K different ITBNs G_k, where $1 \leq k \leq K$ for K different activities, an unknown activity sequence can be classified as one of the K candidate activities by evaluating the likelihood of the model. The model with the highest likelihood, i.e., $\hat{G}_k$, is selected as the most likely activity, i.e.,

$$\hat{G}_k = \arg\max_{G_k} LL(\mathbf{Z}|G_k), \tag{13}$$

where LL denotes log likelihood, $\mathbf{Z} = \{Z_j\}_{j=1}^{n}$ is the set of observation nodes Z_j of X_j. The unknown activity is recognized as one of the candidate activities k whose learned model G_k can best explain the observations with temporally consistent event instances. Notice that ITBNs, in contrast with existing graphical models, use both events and their temporal dependencies over durations of time (instead of points of time) to perform activity modeling and recognition.

Besides offline activity recognition, the ITBN model can be applied to online recognition as well, for example, online abnormal activity detection. Especially, given a known activity, as the video streams in, the model can be used to evaluate the compatibility of already observed events with the model and declares an anomaly if the compatibility is below a threshold. The model can also be used to predict the next event and when it will occur.

6 Experiment with Synthetic Data

The performance of the proposed ITBN model is first evaluated by testing against synthetic data to systematically study its performance under different conditions. To be realistic and to have meaningful events, we propose to use the cargo loading example as the basis for the synthetic experiment. Based on the real scenarios of cargo loading, it includes six events: cargo truck approaching (A), lowering down the cargo lift to the loading position (B), unloading the container to the cargo lift (C), cargo lift lifting up to the unloading position (D), cargo truck leaving (E), and unloading container from the cargo lift (F). In addition, for each event, we define its minimal start time, maximal finish time, and minimal duration, which is shown in Fig. 8. Based on the definition, the six events not only have temporal relations but also maintain certain spatial dependencies (e.g., some events have high probability of co-occurring, while some may be mutually exclusive). For training, 100 cargo loading instances were generated, where the duration of each event was drawn from a uniform distribution on the interval between minimal start time and maximal finish time subject to a minimal duration requirement. In addition, for testing, we generated another set of 100 instances of abnormal cargo loading activity. The abnormal cargo loading instances were generated by violating either the minimal start time or the maximal finish time as defined for each event. Besides ITBN model, it is also necessary to compare ITBN with other competing models. A coupled hidden semi-Markov model (CHSMM) is widely used for modeling interactions among temporal entities. It factors the multiple chains of hidden semi-Markov models (HSMM) [43], [44] so that the HMM has compositional state in both space and variable time duration. It has been demonstrated in [45] that the CHSMM outperforms other HMM variants such as the CHMMs [6], [5], the HSMM [44], and the S-HSMM [11]. Thus, we propose to compare the ITBN with the CHSMM.

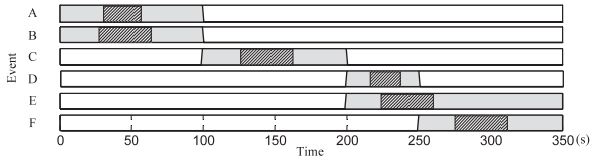


Fig. 8. The minimal start time (the left boundary of the gray area), the maximal finish time (the right boundary of the gray area), and the minimal duration (the shaded area) of the six events are defined for cargo loading activities.

The first dataset of instances of normal cargo loading activities was used to learn ITBN structure and temporal relationships, and to train the CHSMM. The learned ITBN structure and the CHSMM chains are shown in Fig. 9. Notice that the same as the ITBN model, the CHSMM model also contains two layers. The top layer comprises a set of hidden nodes representing the primitive events. The bottom layer comprises a set of observation nodes ingesting event detections for their corresponding event node in the top layer. The structure of the top layer is manually specified according to the domain knowledge of cargo loading.

Then, the testing datasets are used to evaluate the performance of the ITBN model under different conditions. Since the ITBN model requires explicit event detection, the errors with event detection will affect ITBN's performance.

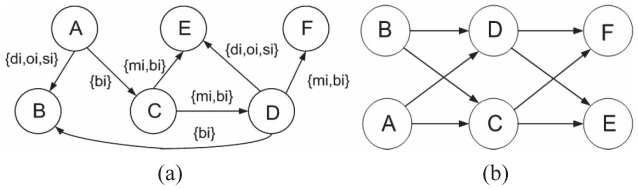


Fig. 9. The learned ITBN structure and temporal relationships (a), and the CHSMM chains (b). The names of the events are abbreviated to: A = cargo truck approaching, B = lowering down the cargo lift to the loading position, C = unloading the container to the cargo lift, D =cargo lift lifting up to the unloading position, E = cargo truck leaving; F = unloading container from the cargo lift. For clarity, the observation nodes are not shown.

We propose to study the performance of the ITBN under different event detection errors, including incorrect event detection and incorrect event time estimation, by comparing with the performance of the CHSMM.

6.1 ITBN Performance under Event Detection Error

Because of the tracking errors or inherent problems with the event detector, one common error with event recognition is misdetection, i.e., the correct event is not detected or is falsely recognized as another event. This experiment studies the performance of ITBN under a varying amount of misdetection rate, i.e., 0 percent (no misdetection), 10 percent, and 20 percent event misdetection, respectively, in sample instances. This is accomplished by perturbing the event labels of the testing data set (both the normal and abnormal loading instances) to simulate incorrect event detection. A fivefold cross validation is performed. Table 6 shows the performance of ITBN and CHSMM under different misdetection rates. It can be seen that both the ITBN and the CHSMM show an excellent performance if all events are correctly detected. As expected, the classification accuracy degrades as the misdetection increases. Fortunately, the ITBN classification performance remains highly stable. For example, when the percentage of misdetected events rises to 20 percent, the CHSMM classification accuracy has a significant drop whereas the ITBN still maintains very high accuracy. It is clear from the table that the CHSMM is highly sensitive to misdetection error, while the ITBN has an overall stable performance and its performance decreases gradually as the misdetection rate increases. This result shows that the ITBN model is more robust to event misdetection compared with the CHSMM.

6.2 ITBN Performance under Event Time Detection Error

Event times (start and finish times) are important to determine the temporal relationships between two events. An automatic event detector often makes mistakes in determining an event's beginning, ending times, as well as the event's duration. In this experiment, we investigate the performance of the ITBN under a varying event time measurement errors. We perturbed the testing data set by perturbing the event start and finish time with a noise varying noise level of ± 10, ± 15, and ± 20 percent of the maximal temporal distance between neighboring events, respectively. To be realistic, a noise of 10 percent is also added to the event misdetection. Table 7 shows the performance of ITBN and CHSMM under different event time errors. It shows again that the ITBN is more robust to time measurement error than the CHSMM.

TABLE 6
Classification Accuracy under Varying Missed Evidence

Event Missed in Samples (%)	Classification Accuracy	
	ITBN	CHSMM
0	1	0.98
10	0.96	0.78
20	0.94	0.59

In summary, the synthetic experiment shows that the proposed ITBNs are robust and sufficient in handling errors resulting from noisy data. Compared with the CHSMM, the ITBN consistently achieves higher performance under varying data noise. The recognition of complex activities with highly unstructured interactions is presented in the next section.

7 Experiment with Real Videos

In this section, we report the activity recognition results using events obtained from real video data. Specifically, the results on the OSUPEL basketball data [26] and the American Football data are discussed. The ITBN model is compared against BN, the DBN, the CHSMM, and the Supervised Latent Dirichlet Allocation topic model (sLDA).

7.1 OSU Basketball Experiments

The OSUPEL basketball data set [26] is publicly available and it consists of multiple players playing against each other on a real basketball court. This dataset is suitable for evaluating detection and localization of multiple primitive events characterized by rich spatiotemporal constraints, as well as complex activities such as different offensive play types that are composed of six primitive events: pass, catch, hold ball, shoot, jump, and dribble. We want to use these six types of primitive events to recognize complex activities. In the two-on-two game videos, we defined two offensive play types as complex activities:

1. Play type 1: Player 1 receives the ball from throw-in and passes to player 2. Player 2 attacks the rim.
2. Play type 2: Player 1 receives the ball from throw-in and attacks the rim directly.

The numbers of the samples for the two offensive play types are 28 and 8, respectively. These two activities both consist of six types of primitive events; they vary in the temporal relationship between them. We believe the ITBN

TABLE 7
Classification Accuracy under Varying Time Error

Time Measurement Error (%)	Classification Accuracy			
	No Events Missed		10% Events Missed	
	ITBN	CHSMM	ITBN	CHSMM
± 10	0.97	0.65	0.84	0.56
± 15	0.83	0.57	0.80	0.55
± 20	0.65	0.56	0.64	0.53

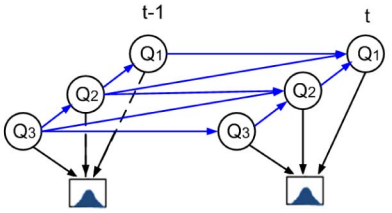


Fig. 10. Generic event DBN model, where the left layer is shown as time slice $t-1$ and the right is time slice t.

model can capture both the spatial relationship and the temporal relationship of the primitive events.

7.1.1 Feature Extraction and Event Recognition

Before discussing the activity recognition results, we first briefly summarize our methods for primitive event detection. The computed tracks of the players in the videos have been already provided in the dataset [26], which are extracted by a template matching-based approach [46]. We extract features from the bounding box of the computed tracks and use a DBN model to detect primitive events.

For feature extraction, we employ two main categories of features: kinematic and image based. The kinematic features are calculated from the track's filtered state estimates, while the image features are calculated from the pixels inside the bounding box of the track's detection. The kinematic features are obtained by filtering the position detections from a given track through a sliding least-squares filter. The least-squares filter produces filtered state estimates that are then used to derive six scene independent kinematic features: speed, heading, change-in-heading, range, entropy of change-in-heading, and curvature. The majority of the image-based features are derived from the standard histogram of oriented gradients [47] and include the gradient magnitude and orientation along with their differences across adjacent frames. Each image feature category also has a corresponding entropy, mean, and standard deviation feature calculated on every frame. The detection's bounding box size and its width-to-height ratio are also included in the image features. An Adaboost feature selector is executed on all 166 dimensions of kinematic and image features in the feature pool, where a decision stump is used as a weak classifier. Since the decision stump chooses the single most discriminative feature that minimizes the overall training error, a count of the selected features results in a ranking of the most discriminative features upon completion of Adaboost. The top 20 most discriminative features are selected on a per event basis and used as the inputs to the observation nodes of the corresponding event model.

For primitive event detection, a DBN is used to model each event. As shown in Fig. 10, the DBN model consists of two layers, a hidden layer and an observation layer. Nodes Q_1, Q_2, and Q_3 are all discrete hidden nodes. Each hidden node represents one of the clustering layers in a hierarchical divisive clustering algorithm [48] whose discrete value corresponds to one of the clusters in that layer. The number of states for a hidden node is the number of clusters in its layer. For each event model, the number of hidden nodes and the number of states for each hidden state are determined experimentally through a cross-validation process, where the optimal structures are chosen by analyzing the probability of correct classification (Pcc). During event detection, a sliding window moves across a querying video and at each location the data in the sliding video is tested against a collection of DBN models, and is assigned the event label that corresponds to the model with the highest likelihood. Based on this label, we can determine which event node shall be instantiated in the ITBN model. Table 8 summarizes the event detection results for the basketball data. The detected events and their temporal interval information are employed as the observation values of the nodes to feed into the ITBN model. Here, we assume that each event occurs at most once in the duration of a complex activity. However, multiple occurrences of the same primitive event may be detected within the activity duration, though it did not happen frequently. If multiple occurrences are detected, we generate single occurrence activity samples based on all the possible combinations of the events to replace the original multiple occurrences activity sample. In each newly generated sample, there is only one occurrence for each event. During training, these samples will be used collectively with other single-instance training samples to train the ITBN model. During testing, all of the single-instance samples produced by a multi-instance query sample will be tested against the ITBN models corresponding to all known activity classes. Then, an average likelihood over all the generated samples is obtained for each activity class, the query will be classified into the activity class with the highest average likelihood.

TABLE 8
Performance of Primitive Event Detection in the OSUPEL Basketball Dataset

	Pass	Catch	Hold	Shoot	Jump	Dribble
RC	0.80	0.80	0.83	0.63	0.61	0.57
FA	0.13	0.30	0.22	0.32	0.47	0.38

Note: RC=Recall, FA=False Alarm

7.1.2 Basketball Activity Recognition

In the activity recognition experiment, we compare our ITBN model with a BN, a DBN, as well as an sLDA model. Specifically, to implement the LDA model, we first followed the same idea in [37], [38] to translate each activity into a bag of spatially distributed optical flow words. The codebook contains the flows that are quantized into eight directions and at positions arranged on a grid with a spacing of 40 pixels. Since the camera of the videos does not remain static, the unexpected flows on the background need to be manually removed. Upon that we employed the supervised LDA [49] to learn the underlying events and perform classification. Ten topics were selected for the experiment.

The experiment was performed with a fivefold cross-validation setting. F1-scores at different classification rates are obtained to demonstrate the performance of each method, where the classified rate is defined as the value of the classified samples divided by the total number of testing samples. The F1-score curves are shown in Fig. 11. We can see that the performance of our ITBN model is significantly better than the other three models at any

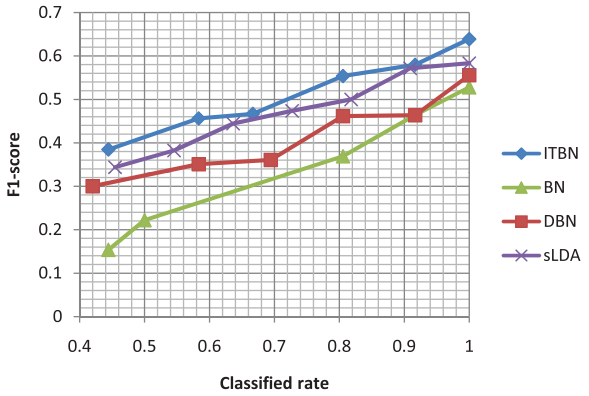


Fig. 11. F1-score curves for basketball play-type classification.

TABLE 9
Basketball Play Type Classification Confusion Matrix

ITBN		BN		DBN		sLDA	
PT1	PT2	PT1	PT2	PT1	PT2	PT1	PT2
0.61	0.39	**0.46**	0.54	**0.61**	0.39	**0.67**	0.33
0.25	**0.75**	0.25	**0.75**	0.62	**0.38**	0.75	**0.25**
Pcc=63.89%		Pcc=52.78%		Pcc=55.56%		Pcc=58.33%	

Note: the probability of correct classification (Pcc) above is average weighted; PT1= play type 1, PT2= play type 2

classification rate. For clarity, we also provide the classification confusion matrices in Table 9. It is clear that ITBN outperforms BN and DBN in recognizing both of the activities. sLDA recognizes play type 1 slightly better than ITBN but much worse for play type 2. Generally, ITBN can perform accurate and robust recognition despite the low event detection performance for some events, as shown in Table 8, mainly due to its ability to take advantage of the rich and complex relationships among events. Besides accuracy, the methods are also different in computational complexity. The sLDA model takes much time to compute flow features and to calculate the visual words, while the feature extraction and event recognition for BN, DBN, and ITBN models can be implemented efficiently.

7.2 American Football Experiments

The American football data are the videos of a Division I college team (Georgia Tech) where the play taxonomy and the tracks were supplied by a former professional football player at Georgia Tech.[1] American football plays are an ideal domain for modeling and recognizing the complex and time-varying temporal relationships (TRs) that exist between multiple-moving objects in coordinated group activities. ITBN also offers a simple, fully representative, and probabilistic tool for modeling the relationships between offensive players actions. For example, each player runs a prescribed route along the football field that is made up of a sequence of events, i.e., a receiver *runs-straight*, *slants in*, and then *receives* the football. At the same time, the quarterback is going through his sequence of events: *step back* (after receiving the snap), *run toward sideline*, and then *throw ball*. The strong TRs between the quarterback and receivers create temporal links between their event nodes in the ITBN. Similar links are learned and created for all TRs that are observed in the data.

1. The football video data were provided courtesy of the Georgia Tech Athletic Association. Video stabilization and tracking were provided by Sima Taheri and Mahesh Ramachandran of the University of Maryland.

TABLE 10
Football Play Types

Id	Name	Brief Description
1	Right	2-3 receivers lined up along the LOS that will block for the running back as he runs to the right of the linemen and down the field
2	Left	2-3 receivers lined up along the LOS that will block for the running back as he runs to the left of the linemen and down the field
3	Middle	2-3 receivers lined up along the LOS that will block for the running back as he runs through a hole that the linemen created in the center of their formation
4	Roll-Out	2-3 receivers, where one or two slant-in while the quarterback (QB) runs toward the line-of-scrimmage
5	Combo	2-3 receivers, where one of them runs-toward-sideline, and the others can run-straight, slant-in, slant-out, or slow-turn-look and turn back for the ball. These occur while the QB steps-back and throws the ball
6	Short	2-3 receivers, where each runs-straight then slow down, turn, and look for the ball. These occur while the QB steps-back and throws the ball.

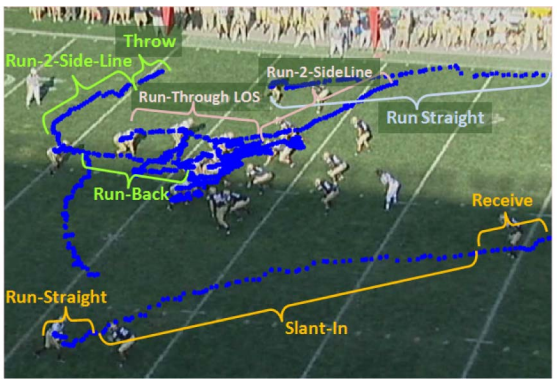


Fig. 12. Frame grab from a *roll-out* play-type example with its labeled events.

Three run play types (*right*, *left*, and *middle*) and three pass play types (*roll-out*, *combo*, and *short*) are considered. Table 10 describes the six play types and the roles of the relevant players. All six play types consist of 2-3 primary receivers and the quarterback. These plays vary in the types of routes that they run, their event types, and the timing and relationships of various events. Fig. 12 shows an example of a *roll-out* play type with the annotated events overlaid on the image; notice the linemen and uninformative offensive players are not included in the event annotations.

7.2.1 Object Tracking and Event Detection

Before tracking, the videos are first preprocessed to stabilize the images to the first video frame. A multiple-hypothesis tracker (MHT) [50] is employed where moving object detections are detected using a temporal variance-based approach [51] within a masked region for the football field, as determined using the expected color of the field. The detection-to-track association uses kinematic state as well as appearance matching cost matrices, where the kinematic states of each track are estimated by a standard Kalman filter, the appearance model is calculated using

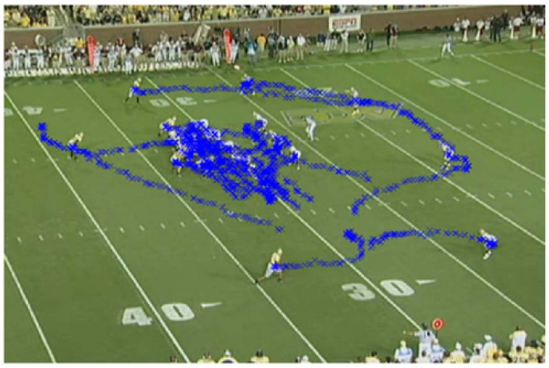


Fig. 13. Example of the right play type with computed tracks overlaid in blue. The offense is on the left and defense on the right.

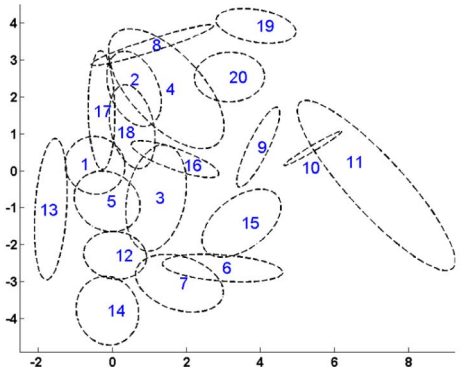


Fig. 14. Twenty 2D Gaussian clusters generated from computed tracks. Each cluster can be chosen as a node in the ITBN.

a Kernel-based color histogram [52] approach that is updated throughout the video. Fig. 13 shows the computed tracks overlaid on the background images.

The tracker's performance was evaluated over all plays using the track completeness factor (TCF) and track fragmentation (TF) metrics from [53], [54]. The mean TCF value is 0.82 with a standard deviation of 0.06, and the mean TF value is 3.28 with a standard deviation of 0.68. TCF measures the proportion of frames in which objects are detected, and TF measures the number of independent tracks that are associated with a given object. A perfect measure for both of these metrics is 1.0. The detection-to-track association performance (TCF) is reasonable, while fragmentation is high. This implies that we cannot assume that objects are correctly tracked through the duration of the activity. Track switching errors are particularly detrimental for activity recognition, as switched tracks can incorrectly indicate impossible or improbable behaviors. The tracker was tuned to avoid switching errors, but this resulted in a higher fragmentation rate.

During primitive event detection, we found that it is difficult to detect the events with explicit semantic meanings based on the computed tracks. Because of the high fragmentation rate of the tracks, we cannot assume getting complete tracks of the players through the duration of the activity. Also, the tracklets cannot preserve the information of the player's ID. Hence, we alternatively use the tracklets to generate a set of statistically distinct multivariate Gaussian clusters. Our goal is to verify whether the ITBN can capture the temporal relationships between not only the explicit events but also the cluster-based events and whether it can maintain accuracy despite difficult tracking.

The clustering is done by performing hierarchical divisive clustering [48] on the features derived from the track's detections, i.e., 2D position estimates. The clustering algorithm starts by assigning all of the detections from all tracks to a single cluster, which is then bifurcated, thus splitting the detections, independent of track Id, into two more clusters. This clustering process continues by bifurcating the cluster with the largest area first, where the area is defined as the product of the covariance's eigenvalues or its determinant. This bifurcation process continues until the desired number of clusters is formed or until the model fit to all the data versus complexity no longer improves. The BIC [40] is used to balance the model's fit to its complexity.

The two sigma boundaries for the final Gaussian clusters are shown in Fig. 14. Based on our observation on the data, the cluster-based events can roughly correspond to certain semantically meaningful football events, as described in Table 11, which is performed by manually associating the semantic events from particular play types with the set of automatically learned clusters. The cluster-based events and their temporal interval information are used as the observation to the ITBN model.

7.2.2 Football Activity Recognition

From the detected cluster events, we can then construct ITBN models for different football play types using the structure learning method described before. Without loss of generality, the ITBN model structure for the *roll-out* play type is shown in Fig. 15. Separate ITBN models are learned for the other play types using the appropriate events from Table 11. Also, the event Ids from Table 11 are listed inside the event nodes of Fig. 15.

The temporal and causal dependencies in Fig. 15 are automatically determined between every pair of nodes i and j using the structure learning method from Section 4.2. After the structure is determined, the parameters are learned from the cluster-based event detections, where a track's evidence vector is the accumulation of all events and temporal evidence from all previous frames for both training and testing. Football play types are classified by testing the unknown play against the library of play type ITBN models and then assigning it the label of the most likely model.

TABLE 11
Cluster-Based Event IDs with Their Descriptions

Event Id	Description
1	Quarterback walks away from the LOS after receiving the snap
13	Quarterback run directly towards side line, parallel to LOS
2,3,4,12,16,18	Pass routes
5	Run through line of scrimmage middle
12, 14	Run through line of scrimmage left side
17	Run through line of scrimmage right side
6	Left receiver run straight deep
7	Left receiver slant-out
8	Right receiver slant-out
9	Middle left receiver slant-in deep
10,11	Clutter tracks of referee or defensive players
15,20	Long pass routes
19	Right receiver run straight deep

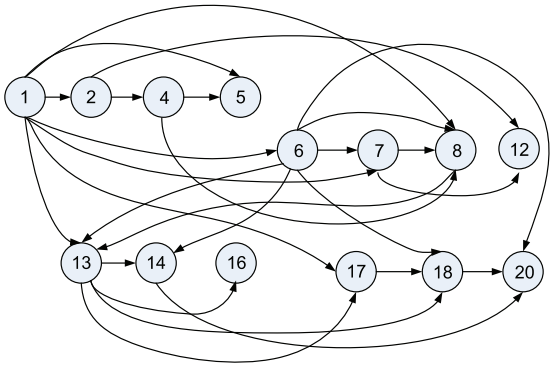


Fig. 15. ITBN for *roll-out* play type. The node Ids correspond to the events in Table 11. For clarity, the observation nodes are not shown in this figure.

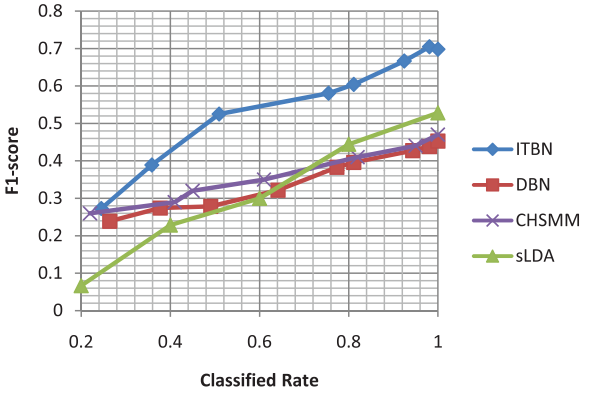


Fig. 16. F1-score curves for football run play-type classification.

For comparison, we propose to compare ITBN with the DBN, sLDA, as well as the CHSMM. In the dataset, the numbers of the three run play types (i.e., right, left, and middle) examples are 11, 23, and 19, respectively. The numbers of the three pass play types (i.e., roll out, combo, and short) examples are 7, 6, and 5, respectively. Because of the large amount of training data required for the topic models, we can only compare with sLDA on the three Run play types. The experiment was evaluated using fivefold cross validation, where the models were learned using 80 percent of the play type examples and tested on the remaining 20 percent. Fig. 16 shows the F1-score curves of the ITBN, DBN, CHSMM, and sLDA methods on the three Run play types. It is clear that ITBN outperforms other competing methods. Table 12 shows the confusion matrix. In Table 12, the average Pcc of ITBN, DBN, CHSMM, and sLDA are 69.81, 45.28, 47.17, and 52.83 percent. It can be seen that ITBN consistently outperforms the other three models in all scenarios, with 20 percent improvement on Pcc over the others. sLDA performs slightly better than DBN and CHSMM on average, but it totally misclassified all the samples in play type "right." The results demonstrate the significance of modeling the temporal relationships by ITBN and its robustness to the smaller training sets. Similar classification performance was obtained for the three pass play types, with the average Pcc of ITBN, DBN, and CHSMM being 44.44, 22.22, and 27.78 percent, respectively.

8 Conclusion

In this paper, we propose the ITBNs that combine the probabilistic semantics of BNs with the temporal semantics of Allen's interval-based framework.

It extends the Allen's IA in two aspects. First, it allows incorporating uncertainties into temporal relationships and into activity inference. Second, it captures not only the temporal relationships but also the spatial relationships among the temporal entities.

The novelty brings about several benefits in modeling a complex activity:

1. The proposed model allows the representation of time-constrained relationships over time intervals, while remaining fully probabilistic and expressive of uncertainties. This new model is more expressive in modeling the parallel and interactive events comprising an activity than its counterparts.
2. The proposed model enables an activity recognition system to effectively use the temporal relationship constraints to compensate for the inaccuracies with event detection, hence improving activity recognition accuracy and robustness.
3. Compared with the existing time-constrained models, our framework not only includes all possible interval temporal relationships, but also incorporates them into a stochastic Bayesian framework to account for uncertainties with these relationships and their measurements.
4. Advanced machine learning methods were proposed to automatically learn both the ITBN model structure and parameters. This is in contrast with most of current methods, which tend to manually specify the model structure and sometimes the parameters as well.
5. Compared with the time-sliced graphical models such as DBNs and CHSMMs on both synthetic and real data, the proposed model can achieve higher performance when modeling complex activities, while also being much more computationally tractable.

From our experiments, we have noticed the strengths of the topic models which use motion features to discover underlying topics, without requiring explicit object tracking and event recognition. Meanwhile, the ITBN is also proven to be powerful enough to capture rich and complex temporal relationships between not only the explicit semantic events but also the implicit cluster-based events and thus with the ability to complement the existing topic models. As part of future work, we are considering

TABLE 12
Run Play Type Classification Confusion Matrix

ITBN			DBN			CHSMM			sLDA		
R	L	M	R	L	M	R	L	M	R	L	M
.36	0	.64	**.27**	.18	.55	**.36**	.45	.19	**0**	.64	.36
0	**.91**	.09	.17	**.48**	.35	.26	**.48**	.26	0	**.78**	.22
.21	.16	**.63**	.05	.42	**.53**	.15	.32	**.53**	0	.53	**.47**
Pcc=69.81%			Pcc=45.28%			Pcc=47.17%			Pcc=52.83%		

Note: the probability of correct classification (Pcc) above is average weighted; R= Right, L= Left, M= Middle

exploiting the topic model idea, i.e., using flow features to automatically identify the underlying topics and formulate the implicit event detection in the form of topic discovery, and then capture the spatial and temporal relationships among topics using the ITBN model. Also, a more powerful temporal relation calculation algorithm will be investigated to relax the single occurrence assumption to be compatible with the situation where multiple occurrences of the same primitive event are detected within the duration of a complex activity.

ACKNOWLEDGMENTS

This work was supported in part by the US Defense Advanced Research Projects Agency under grants HR0011-08-C-0135-S8 and HR0011-10-C-0112. The publication of this paper was supported by the National Natural Science Foundation of China under grant 61202325. Yongmian Zhang and Yifan Zhang contributed equally to this work and should be considered co-first authors.

REFERENCES

[1] P. Turaga, R. Chellappa, V. Subrahmanian, and O. Udrea, "Machine Recognition of Human Activities: A Survey," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 18, no. 11, pp. 1473-1488, Nov. 2008.

[2] C. Pinhanez, "Representation and Recognition of Action in Interactive Spaces," PhD thesis, MIT Media Lab, 1999.

[3] J. Pearl, *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, 1988.

[4] J.F. Allen and G. Ferguson, "Actions and Events in Temporal Logic," *J. Logic and Computation,* vol. 4, no. 5, pp. 531-579, 1994.

[5] N.M. Oliver, B. Rosario, and A.P. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 831-843, Aug. 2000.

[6] M. Brand, N. Oliver, and A. Pentland, "Coupled Hidden Markov Models for Complex Action Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1997.

[7] S. Park and J.K. Aggarwal, "A Hierarchical Bayesian Network for Event Recognition of Human Actions and Interactions," *Multimedia Systems,* vol. 10, no. 2, pp. 164-179, 2004.

[8] R. Hamid, Y. Huang, and I. Essa, "ARGMode Activity Recognition Using Graphical Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2003.

[9] T. Xiang and S. Gong, "Beyond Tracking: Modeling Activity and Understanding Behaviour," *Int'l J. Computer Vision,* vol. 67, no. 1, pp. 21-51, 2006.

[10] S. Gong and T. Xiang, "Recognition of Group Activities Using Dynamic Probabilistic Networks," *Proc. IEEE Int'l Conf. Computer Vision,* 2003.

[11] T.V. Duong, H.H. Bui, D.Q. Phung, and S. Venkatesh, "Activity Recognition and Abnormality Detection with the Switching Hidden Semi-Markov Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2005.

[12] Y. Shi, A.F. Bobick, and I.A. Essa, "Learning Temporal Sequence Model from Partially Labeled Data," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 1631-1638, 2006.

[13] A. Fernadez-Leal, V. Moret-Bonillo, and E. Mosqueira-Rey, "Causal Temporal Constraint Networks for Representing Temporal Knowledge," *Expert Systems with Applications,* vol. 36, no. 2009, pp. 27-42, 2009.

[14] R. Nevatia, T. Zhao, and S. Hongeng, "Hierarchical Language-Based Representation of Events in Video Streams," *Proc. Second IEEE Workshop Event Mining: Detection and Recognition of Events in Video,* 2003.

[15] A. Hakeem, Y. Sheikh, and M. Shah, "CASE: A Hierachical Event Representation for the Analysis of Videos," *Proc. 19th Nat'l Conf. Artificial Intelligence,* 2004.

[16] F. Fusier, V. Valentin, F. Bremond, M. Thonnat, M. Borg, D. Thirde, and J. Ferryman, "Video Understanding for Complex Activity Recognition," *Machine Vision and Applications,* vol. 2007, no. 18, pp. 167-188, 2007.

[17] S. Hongeng, R. Nevatia, and F. Bremond, "Video-Based Event Recognition: Activity Representation and Probabilistic Recognition Methods," *Computer Vision and Image Understanding,* vol. 96, no. 2, pp. 129-162, 2004.

[18] M.S. Ryoo and J.K. Aggarwal, "Spatio-Temporal Relationship Match: Video Structure Comparison for Recognition of Complex Human Activities," *Proc. IEEE Int'l Conf. Computer Vision,* 2009.

[19] C.F. Aliferis and G.F. Cooper, "A Structurally and Temporally Extended Bayesian Belief Network Model: Definitions, Properties, and Modeling Techniques," *Proc. 12th Ann. Conf. Uncertainty in Artificial Intelligence,* 1996.

[20] E. Santos Jr. and J.D. Young, "Probabilistic Temporal Networks: A Unified Framework for Reasoning with Time and Uncertainty," *Int'l J. Approximate Reasoning,* vol. 20, pp. 263-291, 1999.

[21] J.D. Young and E. Santos Jr, "Introduction to Temporal Bayesian Networks," *Proc. Seventh Midwest AI and Cognitive Science Conf.,* 1996.

[22] S.S. Intille and A.F. Bobick, "Recognizing Planned, Multiperson Action," *Computer Vision and Image Understanding,* vol. 81, pp. 414-445, 2001.

[23] E. Santos Jr., "On the Generation of Alternative Explanations with Implications for Belief Revision," *Proc. Seventh Conf. Uncertainty in Artificial Intelligence,* pp. 339-347, 1991.

[24] B. Milch and S. Russell, "First-Order Probabilistic Languages: Into the Unknown," *Proc. 16th Int'l Conf. Inductive Logic Programming,* pp. 10-24, 2007.

[25] V.I. Morariu and L.S. Davis, "Multi-Agent Event Recognition in Structured Scenarios," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 3289-3296, 2011.

[26] W. Brendel, A. Fern, and S. Todorovic, "Probabilistic Event Logic for Interval-Based Event Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 3329-3336, 2011.

[27] S. Sanghai, P. Domingos, and D. Weld, "Relational Dynamic Bayesian Networks," *J. Artificial Intelligence Research,* vol. 24, no. 2005, pp. 759-797, 2005.

[28] B. Milch, B. Marthi, S. Russell, D. Sontag, D.L. Ong, and A. Kolobov, "BLOG: Probabilistic Models with Unknown Objects," *Proc. Int'l Joint Conf. Artificial Intelligence,* pp. 1352-1359, 2005.

[29] M. Richardson and P. Domingos, "Markov Logic Networks," *Machine Learning,* vol. 62, pp. 107-136, Feb. 2006.

[30] J.M. Siskind, "Grounding the Lexical Semantics of Verbs in Visual Perception Using Force Dynamics and Event Logic," *J. Artificial Intelligence Research,* vol. 15, pp. 31-90, 2001.

[31] M. Albanese, R. Chellappa, V. Moscato, A. Picariello, V.S. Subrahmanian, and P. Turaga, "A Constrained Probabilistic Petri Net Framework for Human Activity Detection in Video," *IEEE Trans. Multimedia,* vol. 10, no. 8, pp. 1429-1443, Dec. 2008.

[32] R. Hamid, S. Maddi, A. Bobick, and M. Essa, "Structure from Statistics—Unsupervised Activity Analysis Using Suffix Trees," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[33] M.S. Ryoo and J.K. Aggarwal, "Semantic Representation and Recognition of Continued and Recursive Human Activities," *Int'l J. Computer Vision,* vol. 2009, no. 82, pp. 1-24, 2009.

[34] Y.A. Ivanov and A.F. Bobick, "Recognition of Visual Activities and Interactions by Stochastic Parsing," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 852-871, Aug. 2000.

[35] A. Hakeem and M. Shah, "Learning, Detection and Representation of Multi-Agent Events in Videos," *Artificial Intelligence,* vol. 71, nos. 8/9, pp. 586-605, 2007.

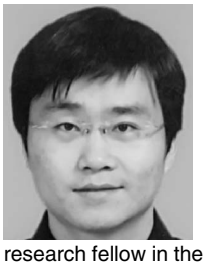
[36] A. Gupta, P. Srinivasan, J. Shi, and L.S. Davis, "Understanding Videos, Constructing Plots—Learning a Visually Grounded Storyline Model from Annotated Videos," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[37] D. Kuettel, M. Breitenstein, L.V. Gool, and V. Ferrari, "Whats Going On? Discovering Spatio-Temporal Dependencies in Dynamic Scenes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.

[38] T. Hospedales, J. Li, S. Gong, and T. Xiang, "Identifying Rare and Subtle Behaviors: A Weakly Supervised Joint Topic Model," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 33, no. 12, pp. 2451-2464, Dec. 2011.

[39] J.F. Allen, "Maintaining Knowledge about Temporal Intervals," *Comm. ACM*, vol. 26, no. 11, pp. 832-843, 1983.
[40] G. Schwarz, "Estimating the Dimension of a Model," *Annals of Statistics*, vol. 6, pp. 461-464, 1978.
[41] C.P. de Campos and Q. Ji, "Efficient Structure Learning of Bayesian Networks Using Constraints," *J. Machine Learning Research*, vol. 12, pp. 663-689, 2011.
[42] D.G.D. Hecherman and D.M. Chickering, "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data," *Machine Learning*, vol. 20, pp. 197-243, 1995.
[43] J.D. Ferguson, "Variable Duration Models From Speech," *Proc. Symp. Application Hidden Markov Models Text Speech*, 1980.
[44] C. Mitchell, M. Harper, and L. Jamieson, "On the Complexity of Explicit Duration HMMs," *IEEE Trans. Speech and Audio Processing*, vol. 3, no. 3, pp. 213-217, May 1995.
[45] P. Natarajan and R. Nevatia, "Coupled Hidden Semi Markov Models for Activity Recognition," *Proc. IEEE Workshop Motion and Video Computing*, 2007.
[46] F. Jurie and M. Dhome, "Real Time Robust Template Matching," *Proc. British Machine Vision Conf.*, 2002.
[47] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 886-893, 2005.
[48] A. Guenoche, P. Hansen, and B. Jaumard, "Efficient Algorithms for Divisive Hierarchical Clustering with Diameter Criterion," *J. Classification*, vol. 8, pp. 5-30, 1991.
[49] C. Wang, D. Blei, and F.-F. Li, "Simultaneous Image Classification and Annotation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1903-1910, June 2009.
[50] D. Reid, "An Algorithm for Tracking Multiple Targets," *IEEE Trans. Automatic Control*, vol. 24, no. 6, pp. 843-854, Dec. 1979.
[51] S. Joo and Q. Zheng, "A Temporal Variance-Based Moving Target Detector," *Proc. IEEE Int'l Workshop Performance Evaluation of Tracking and Surveillance*, Jan. 2005.
[52] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564-577, May 2003.
[53] A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, and W. Hu, "Multi-Object Tracking through Simultaneous Long Occlusions and Split-Merge Conditions," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006.
[54] K. Smith, D. Gatica-Perez, J. Odobez, and B. Sileye, "Evaluating Multi-Object Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005.

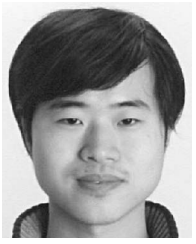
Yongmian Zhang received the PhD degree in computer engineering from the University of Nevada-Reno in 2004. He is currently a senior research scientist at Konica Minolta Laboratory U.S.A. He also held a research position in the Department of Electrical, Computer, and Systems Engineering at Rensselaer Polytechnic Institute, and was a research scientist with several industrial companies focusing on video surveillance systems. His research interests include computer vision, video processing, probabilistic graphical models, affective computing, and gesture-based human-computer interaction. He is a member of the IEEE.



Yifan Zhang received the BE degree in automation from Southeast University in 2004 and the PhD degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences in 2010. Then, he joined the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, where he is currently an assistant professor. From 2011 to 2012, he was a postdoctoral research fellow in the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, New York. His research interests include probabilistic graphical models, activity recognition, and video semantic analysis. He is a member of the IEEE.

Eran Swears received the BS degree in electrical engineering (EE) from Rensselaer Polytechnic Institute (RPI) in 2001, after which he joined the Discrimination Group at Lockheed Martin in Moorestown, New Jersey, where he researched and developed algorithms for intercontinental ballistic missile trackers. In parallel, he received the MS degree in electrical engineering from Drexel University in 2005 and progressed to Tracker Team lead. In 2006, he joined the Computer Vision Group at GE Global Research as a contractor, where he researched motion pattern learning and anomaly detection in video. He has been employed by Kitware, Inc., since 2007 as a member of the Computer Vision research staff, where he has been the principle researcher or project lead on several US Defense Advanced Research Projects Agency efforts. He is currently working toward the PhD degree in electrical engineering at RPI. His research interests include computer vision, pattern recognition, machine learning, and, in particular, activity modeling and recognition using graphical models and probabilistic logic. He is a member of the IEEE.

Natalia Larios received the BS degree in computer engineering from UNAM in Mexico in 2003 and graduated from the University of Washington with the MS and PhD degrees in electrical engineering in 2010. She is a researcher at Microsoft doing safety research in automated account abuse and compromise detection based on machine learning models and user behavior features. She was a postdoctoral research associate during 2011 at Rensslaer Polytechnic Institute in Troy, New York, while participating in this project. Her interests include activity detection, object recognition, and image classification employing machine learning and probabilistic modeling. She is a member of the IEEE.



Ziheng Wang received the BS degree in electrical engineering from Tsinghua University in 2010. He is currently working toward the PhD degree in electrical, computer, and systems engineering at Rensselaer Polytechnic Institute. His research interests include machine learning, pattern recognition, computer vision, and graphical models. He is a student member of the IEEE.

Qiang Ji received the PhD degree in electrical engineering from the University of Washington. He is currently a professor in the Department of Electrical, Computer, and Systems Engineering at Rensselaer Polytechnic Institute (RPI). He recently served as a program director at the US National Science Foundation (NSF), where he managed NSF's computer vision and machine learning programs. He also held teaching and research positions with the Beckman Institute at the University of Illinois at Urbana-Champaign, the Robotics Institute at Carnegie Mellon University, the Department of Computer Science at the University of Nevada at Reno, and the US Air Force Research Laboratory. He currently serves as the director of the Intelligent Systems Laboratory at RPI. His research interests are in computer vision, probabilistic graphical models, information fusion, and their applications in various fields. He has published more than 160 papers in peer-reviewed journals and conferences. His research has been supported by major governmental agencies including NSF, NIH, Defense Advanced Research Projects Agency, ONR, ARO, and AFOSR as well as by major companies, including Honda and Boeing. He is an editor on several related IEEE and international journals and he has served as a general chair, program chair, technical area chair, and program committee member for numerous international conferences/workshops. He is a senior member of the IEEE and a fellow of the IAPR.