A General Framework for Recognizing Complex Events in Markov Logic

Young Chol Song, Henry Kautz, Yuncheng Li, Jiebo Luo
University of Rochester
Department of Computer Science
Rochester, NY USA
{ysong,kautz,yli,jluo}@cs.rochester.edu

Abstract

We present a robust framework for complex event recognition that is well-suited for integrating information that varies widely in detail and granularity. Consider the scenario of an agent in an instrumented space performing a complex task while describing what he is doing in a natural manner. The system takes in a variety of information, including objects and gestures recognized by RGB-D and descriptions of events extracted from recognized and parsed speech. The system outputs a complete reconstruction of the agent's plan, explaining actions in terms of more complex activities and filling in unobserved but necessary events. We show how to use Markov Logic (a probabilistic extension to first-order logic) to create a theory in which observations can be partial, noisy, and refer to future or temporally ambiguous events; complex events are composed from simpler events in a manner that exposes their structure for inference and learning; and uncertainty is handled in a sound probabilistic manner. We demonstrate the effectiveness of the approach for tracking cooking plans in the presence of noisy and incomplete observations.

Introduction

Consider a situation where you are observing a person demonstrating both physically and verbally how to perform a complex task, for example, preparing a cup of tea. The subject performs simple actions (*e.g.*, picking up a tea kettle), which are part of more complex activities (filling the kettle from the sink), and which in turn are part of yet higher-level activities (preparing hot water), *ad infinitum*. Actions are inextricably connected to changes in the state of the world (moving the cup changes its location), even if the change is not directly observable (stirring the tea after adding sugar dissolves the sugar). The subject may refer to actions in the past ("I've finished filling the kettle..."), the current moment ("The water is heating..."), or in the future ("I still have to get the milk and sugar..."), and complex events can overlap temporally (the subject fetches the tea box while the water is heating). The subject may describe an event at different levels of abstraction ("I'll heat heat water" *vs* "I'll heat water in the microwave"), or provide a partial verbal description, which is resolved by context ("Now I pour the water [from the kettle into the cup]"). Similarly, visual percepts of events may be incomplete due to visual resolution or obscuring objects, and only disambiguated by context (hand removes *something* from tea box).

A human observer reflexively tries to understand the situation by explaining what he sees and hears in terms of the subject's *plan*: a coherent, connected structure of observed, hypothesized, and predicted structure of actions and properties. When the subject is a teacher, the latter must piece together a new plan. In other cases, the plan is one familar to the observer, whose task becomes identifying, instantiating, and tracking the plan; such is the case, *e.g.*, when a teacher observes a student at work. For this thought exercise, we focused on cooking, but the same principles apply to many domains where there is a practical need for automated plan recognition, such as wet labs, medical procedures, equipment maintenance, and surveillance.

While there is a rich history of research on plan recognition (briefly recapped in the next section), most work makes assumptions about the nature of actions and observations that are violated by the simple example above. We argue that a general framework for plan recognition should meet the following criteria: (i) Be robust across variations in the appearance of a scene and the language used to describe it: *i.e.*, provide a semantic as opposed to an appearance model. (ii) Support easy knowledge engineering, *e.g.*, for defining events in terms of changes of properties of objects and/or collections of other events. (iii) Represent both decomposition and abstraction event hierarchies, with no fixed number of levels. (iv) Treat instances of events as entities to which reference can be made: *e.g.*, support event reification. (v) Allow events that are not temporally disjoint, and observations that arrive out of temporal order.

The contributions of this paper include defining and implementing a framework meeting these criteria based on Markov Logic, a knowledge representation and reasoning system that combines first-order logic with probabilistic semantics. Our implementation includes a capable vision system for tracking the state of objects using an RGB-D (Kinect) camera together with an uncalibrated high-definition camera to increase accuracy. Low-level actions are defined in terms of qualitative spatial and temporal relations rather than visual appearance, so the system does not need to be trained on particular environments. We leverage

a domain independent natural language parser to extract action descriptions and temporal contraints from the subject's narration. Our experiments demonstate accurate recognition and tracking of complex plans, even as visual inputs to the system are purposefully degraded. Finally, we briefly describe how our future work on learning from demonstration builds upon this framework.

Background & Related Work

Our project builds upon work from a wide variety of fields: machine learning, knowledge representation, pervasive computing, computer vision, and computational linguistics. We provide a brief overview of only the most direct precedents.

Markov Logic [Richardson and Domingos, 2006] is a language for representing both logical and probabilistic information in the form of weighted logical formulas. Formulas that include quantified variables are taken to represent the set of ground formulas that can be formed by replacing the variables with constants. The probability of a possible world is proportional to the exponentiated sum of the weights of the ground formulas that are true in that world. The task of finding a most likely explanation of a set of data becomes maximum weighted satisfiability, and can be solved by local search or backtracking methods (*e.g.*, [Ansótegui, Bonet, and Levy, 2013]).

Plan recognition was identified as a core reasoning problem in early research in AI and cognitive science [Schmidt, Sridharan, and Goodson, 1978]. Kautz (1991) developed a logical framework for plan recognition that met the criteria of expressiveness for action abstraction, decomposition, reification, and temporal generality, but did not handle probabilistic information and was never applied to observations from sensor data. The Markov Logic framework for plan recognition by Singla and Mooney (2011) handled probabilities, but was limited to a two-level hierarchy, did not reify actions, and was also never applied to sensor data.

Several groups explored Markov Logic for activity recognition in video [Tran and Davis, 2008; Kembhavi, Yeh, and Davis, 2010; Morariu and Davis, 2011], but did not consider multi-level hierarchies and employed *ad hoc*rules for inferring unobserved events. Of these, Morariu et al. (2011) is closest to our framework, in that it associated events with time intervals rather than time points.

Lei et al. (2012) demonstrated robust tracking of low-level kitchen objects and activities (*e.g.*, pour, mix, *etc.*) using a consumer Microsoft Kinect-style depth camera (RGB-D). Their approach is similar to ours for low-level action recognition, but differs in that they inferred actions from object constraints and appearance-based motion flow, while we use object constraints and relative qualitative spatial position.

We employ the non-domain specific TRIPS parser [Allen, Swift, and de Beaumont, 2008] to extract action descriptions from narration. There is growing interest in machine learning and computational linguistics in models that unify visual perception and natural language processing. This includes using language to supervise machine vision (*e.g.*, [Gupta and Mooney, 2010]) and simultaneous learning of visual and linguistic attributes (color, shape, *etc.*) [Matuszek et al., 2012].

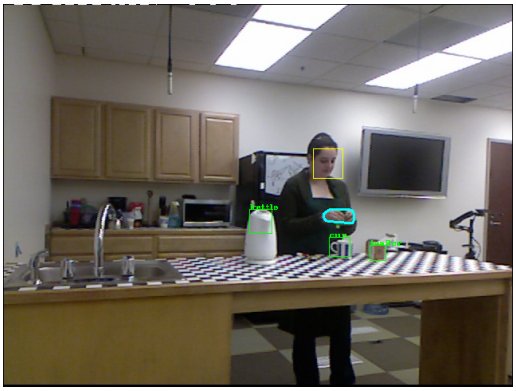

Figure 1: An example frame generated by the vision subsystem. The target objects are in green bounding boxes, the agent's face is in a yellow bounding box, and her hands are in cyan bounding boxes.

The grounded language approach of Tellex et al. (2011), like ours, integrates visual, linguistic, and background knowledge in a general probabilistic model, but has not yet considered plans or complex actions.

Other general formalisms that have been developed for representing and reasoning about hierarchically structured actions include stochastic grammars [Moore and Essa, 2001] and hierarchical hidden Markov models [Bui, 2003; Natarajan et al., 2008]; however, grammars have difficulty representing non-disjoint actions, and the HMM models also fix the number of levels in the hierarchy. Event logic [Brendel, Fern, and Todorovic, 2011] provides a compact notation for probabilistic models relating interval-based actions and properties. Propagation networks [Shi et al., 2004] use partially-order plans to encode the transition relationship in a dynamic Bayesian network.

Detecting Low-Level Events

To facilitate high-level inference, it is important to detect and track the interactions between the agent and objects relevant to the demonstration from sensory data. To that end, we build a computer vision-based subsystem to detect low-level events.

Visual Hand and Object Detection and Tracking

In order to detect the agent's face, hands and the objects interacting with the hands, a robust vision subsystem is constructed to incorporate RGB-D information collected from the Kinect. The vision subsystem starts with detecting skin area based on color information. With the detected skin area, the agent's hand is detected using discriminative connected components analysis. By taking advantage of several common observations, the object interacting with the hand is extracted based on hand detection and depth information. Given a sequence of demonstrations, the vision subsystem keeps track of a small set of predefined target objects aided by temporal smoothing of a MeanShift-based tracking algorithm [Comaniciu, Ramesh, and Meer, 2003]. Following is a

brief summary of the methods employed:

Skin modeling In order to make the vision subsystem adaptive to different lighting conditions, an *image-specific* Gaussian Mixture Model (GMM) is fitted over the pixels inside the detected face bounding box. Face detection is accomplished per frame according to [Viola and Jones, 2004]. We assume that the majority of the area inside the detected face represents skin, which corresponds to the largest cluster in the fitted GMM. For a pixel outside the face bounding box, the Mahalanobis distance to the largest GMM component is computed as a skin score. In order to transform this real-valued score into a binary decision value, a two-parameter sigmoid classifier similar to Platt scaling in SVM (support vector machine) is trained on the fly.

Discriminative hand detection A discriminative Connected Components (CC) analysis is performed over the skin area binary map using SVM. For each CC in the skin area binary map, the following features are used:

- normalized CC size;
- normalized spatial distance to the detected face;
- width-height ratio of the CC bounding box;
- histogram of oriented gradients (HOG) [Dalal and Triggs, 2005];
- distance to the face area.

Hand to object detection Taking advantage of several common observations, we extract the objects interacting with the hands by segmenting regions in the point cloud that are close to the hands but not part of the hands or body.

Multi-object tracking Since there are occlusion and perspective variations from time to time during the demonstration, object detection cannot be expected to be perfect. A multi-object tracking module is constructed to enforce temporal smoothing, particularly compensating for the missed detections. There are two key operations in a multi-object tracking system, tracking and association. MeanShift-based tracking is used for frame to frame object tracking and color histogram distance is used as a matching score in the common Hungarian Algorithm to associate tracking and detections [Blackman, 1986].

Object identification With the multi-object tracking module, the objects interacting with the hand are identified. Since there are limited number of objects of interest, we use a predefined set of named target models to perform object identification. Basically, the visual distance between each target object and each tracked object is computed, and if a good match is found, the location of the target object is updated accordingly. For our prototype, we use simple color histogram distance as a visual matching score, although it can be extended to more sophisticated models.

In summary, the video processing steps constructed in this vision subsystem are able to adequately detect the agent's face, hands and the objects interacting with the hands, which are supplied as low level features to infer atomic events. Precision for hand and object tracking and object identification was 0.84 and recall was 0.89 on our test videos. Figure 1 shows example results of the vision subsystem, where hands and objects are detected, tracked and labeled.

Single Object	Orientation	Straight, Tilted, Upside-down
	Motion	Stationary, In-motion
	Location	Counter, Sink, Cupboard
Object Pair	Relative Location	Above, Directly-above, Co-planar Below, Directly-below
	Distance	Adjacent, Near, Far
Object-Hand	Relation	Held, Not-held,
Subject	Location	Counter, Sink, Cupboard

Table 1: Set of visual fluents used in our "making tea" scenarios.

Grasp	Begins	Fluent(Object, Hand, Held)
	Ends	Fluent(Object, Motion, In-motion)
	Holds	Fluent(Object, Motion, Stationary)
Release	Begins	Fluent(Object, Motion, Stationary)
	Ends	Fluent(Object, Hand, Not-held)
	Holds	Fluent(Object, Hand, Held)
Pour	Begins	Fluent(O1, Orientation, Tilted)
	Ends	Fluent(O1, Orientation, Straight)
	Holds	Fluent(O1, O2, RelLoc, D-Above) Fluent(O1, Hand, Held)

Table 2: Examples of low-level events defined by sets of fluents.

Low-Level Event Generation

The hands and objects identified using the vision subsystem are translated into discrete visual fluents. These are visible qualitative features of the scene that span over intervals throughout the session, which include subject location, qualitative object-object and object-hand relations. A set of visual fluents frequently used in our system is shown in Table 1. These fluents, while being able to represent relevant aspects of the scene, are also represented in a setting-independent way, so that a set of low-level fluents in one scene can be transferred to another without having to re-train the vision subsystem.

Low-level events are generated by the existence of one or more visual fluents. Some fluents or changes in fluents may trigger the start or end of a particular event, while other fluents must hold over the entire event. For example, a *Pour* event is triggered by an object tilting while it is directly above another object and the object in question is in the hand of the subject, while a *Release* is triggered by a change in movement while the object is being released. Each low-level event is described in terms of visual fluents. Some events are

$\text{Dabstracts}(t_1, t_2)$ Event type t_1 abstracts type t_2.

$\text{Dpart}(t_1, p, t_2)$ Events of type t_1 include a part p of type t_2.

$\text{DrelEP}(t, r, p)$ The temporal relation r holds between any event of type t and its part p.

$\text{DrelPP}(t, r, p_1, p_2)$ The temporal relation r holds between parts p_1 and p_2 of any instance of t.

$\text{Occurs}(t, e)$ An event instance e of type t occurs.

$\text{Part}(e_1, pa2)$ Event instance e_1 includes instance e_2 as a part p.

$\text{Rel}(e_1, r, e_2)$ The temporal interval relation r holds between event instances e_1 and e_2.

$\text{Stime}(e, n)$ Event instance e starts at the integer-valued moment n.

$\text{Etime}(e, n)$ Event instance e ends at the integer-valued moment n.

Table 3: Core predicates in the event theory. Predicates that begin with "D" are used to define a domain, while `Occurs`, `Part`, and `Rel` hold for particular event instances.

defined in Table 2.

Representing and Reasoning About Complex Events

We now describe a Markov Logic theory that meets our criteria for plan recognition. Throughout this section we will use the general term "event", rather than action or plan. We begin by introducing predicates that define the sets of event types, event instances, and abstraction and decomposition relationships between events. Any particular domain is specified by defining the domains of these predicates. We then define *generic* axioms for predicting future and unobserved events on the basis of ongoing complex events, and abductively infer complex events from observations of subevents. This approach simplifies domain-specific knowledge engineering, and (in future work) turns the task of learning new events into learning the extent of the definitional predicates, rather than the unconstrained problem of learning arbitrary logical formulas. Our implementation uses the implementation of Markov Logic called "Tuffy" [Niu et al., 2011]. Tuffy extends first-order syntax with scoping and datalog rules, which our implementation makes use of to substantially improve performance. Tuffy also restricts Markov Logic syntax by requiring that each formula be equivalent to a single clause. In order to keep this section brief and clear, howevever, we present logically equivalent axioms in pure Markov Logic.

Table 3 lists the predicates used to define a domain and to describe a particular situation in terms of the events that actually occur. Instances of events are reified, that is, are represented as individuals. Event types are organized into a hierarchy; an instance of a type is also an instance of all abstractions of the type. By default, an event of a given type is also an instance of some known specialization of the type. This is expressed by a weighted (soft) axiom. The weights for soft rules can be learned from examples or estimated manually; in the experiments reported in this paper, estimated weights were sufficient. The axioms are thus:

$$\text{Dabstracts}(t_1, t_2) \wedge \text{Occurs}(e, t_2) \Rightarrow \text{Occurs}(e, t_1).$$

$$10\ \ \text{Occurs}(e, t_1) \Rightarrow \exists\, t_2\ \ \text{Dabstracts}(t_1, t_2) \wedge \text{Occurs}(e, t_2)$$

Temporal relationships between events are expressed using Allen's interval algebra [Allen, 1983], where event instances are treated as intervals. An integer timestamp can optionally be associated with the start and/or end time of an event. The intended semantics is captured by two sets of axioms, the first involving interval relations and endpoints, and the second involving triples of interval relations. An example of the first sort assert that if two events (intervals) meet, the end point of the first must equal the start point of the second; an example of the second is that "begun by" is transitive:

$$\text{Meets}(e_1, e_2) \wedge \text{Etime}(e_1, n_1) \Rightarrow \text{Stime}(e_2, n_2).$$

$$\text{Rel}(e_1, \text{BegunBy}, e_2) \wedge \text{Rel}(e_2, \text{BegunBy}, e_3) \Rightarrow \text{Rel}(e_1, \text{BegunBy}, e3).$$

For example, the formula

$$\text{Occurs}(\text{BoilWater}, \text{E}_1) \wedge \text{Part}(\text{E}_1, \text{Step}_1, \text{E}_2) \wedge \text{Occurs}(\text{FillKettle}, \text{E}_2) \wedge \text{Rel}(\text{E}_1, \text{BegunBy}, \text{E}_2) \wedge \text{Stime}(\text{E}_2, 109).$$

asserts that an instance of the complex event boiling water occurs, and that it is begun by the sub-event of filling a kettle. The filling starts at time 109. As a consequence of the general temporal axioms, the boiling water event also starts at time 109; both events end at unspecified times greater than 109.

Distinct from the abstraction hierarchy is a decomposition, or part-of, hierarchy. There are three types of axioms for complex events. The *prediction* axiom assert that if a complex event occurs, each of its parts occurs by default.

$$10\ \ \text{Occurs}(t_1, e_1) \wedge \text{Dpart}(t_1, p, t_2) \Rightarrow \exists\, e_2\, \text{Occurs}(t_2, e_2) \wedge \text{Part}(e_1, p, e_2)$$

The *constraint* axioms assert that the defined temporal constraints among a complex event and its parts are satisfied.

$$\text{DrelEP}(t, r, p) \wedge \text{Occurs}(t, e) \wedge \text{Occurs}(t_1, e_1) \wedge \text{Part}(e, p, e_1) \Rightarrow \text{Rel}(e, r, e_1).$$

$$\text{DrelPP}(t, r, p_1, p_2) \wedge \text{Occurs}(t, e) \wedge \text{Occurs}(t_1, e_1) \wedge \text{Occurs}(t_2, e_2) \wedge$$

$$\text{Part}(e, p1, e_1) \wedge \mathit{Part}(e, p2, e_2) \Rightarrow \text{Rel}(e_1, r, e_2).$$

Finally, *abduction* axioms allow complex events to be inferred on the basis of their parts. These axioms state that by

default an event is part of a more complex event:

$$10\ \ \mathrm{Occurs}(t_1, e_1) \Rightarrow \\ \exists\, t_2\ e_2\ p \\ \mathrm{Dpart}(t_2, p, t_1) \wedge \\ \mathrm{Occurs}(t_2, e_2) \wedge \\ \mathrm{Part}(e_2, p, e_1)$$

An observer should prefer more likely explanations and should not assume events occur without evidence. These preferences are captured by encoding a prior probability over the occurrance of events of various types by negative weighted clauses. For example,

$$-1\ \ \mathrm{Occurs}(\mathrm{MakeTea}, e)$$

$$-2\ \ \mathrm{Occurs}(\mathrm{MakeCocoa}, e)$$

indicates that prior odds ratio of making tea to making coffee is e^{-1}/e^{-2}.

Experiments

We tested our framework on a multi-modal corpus we collected of people preparing beverages in an instrumented kitchen [Swift et al., 2012]. In each of the sessions, participants were asked to conduct the activity and at the same time verbally describe what they were doing. An RGB-Depth sensor, HD video camera, and lavalier and ceiling microphones were used for data collection.

For the ground truth, activities in the sessions were manually annotated by observing recorded videos performed by the participants. Each low-level event was annotated with an action (*e.g.*, grasp, carry, open) and attributes, such as objects (*e.g.*, cup, kettle, teabox) and paths (*e.g.*, to, from, into).

Inferring Missing Events

We axiomatized the events (actions) that occurred in making tea into a multi level hierarchy. The domain includes low-level events such as "open kettle", middle-level complex events such as "boil water", and top-level events such as "make tea". Table 4 lists the event types involved in making tea. Our plan library included other high-level events, such as "make cocoa", that shared low-level events with "make tea". The "boil water" event abstracted two more specialized events: boiling water using an electric kettle, and boiling water using a microwave.

For our initial set of experiments, we considered the task of inferring all of the events that occurred in the scenario on the basis of a sparse set of observed low-level events. Table 5 shows the results on the "making tea" scenarios as the amount of evidence varied from 100% of the low-level events to no evidence. For each set of observations, MPE (most probable explanation) inference was performed 10 times, and the average percent of all the events that actually occurred calculated. Averages were taken across runs of MPE and scenarios. This shows that despite a shortage of low-level event evidence, our framework allows us to reconstruct a significant porition of the plan using our hierarchy.

Top	Middle Level	Low Level
Make Tea	FillKettle	GraspKettle, CarryKettle, TurnonFaucet, FillWater TurnoffFaucet, CarryKettle ReleaseKettle
	GetIngredients	GoToCupboard, GetCupFromCupboard GetTeaboxFromCupboard
	PrepareTeabag	GraspTeabox, OpenTeabox, PutTeabagIntoCup
	BoilWater	TurnOnKettle, TurnOffKettle
	PourHotWater	GraspKettle, PourWaterIntoCup ReleaseKettle

Table 4: Events in the "make tea" scenario.

% Low-level Obs.	100	80	60	40	20	0
% Events Inferred	100	87	69	84	44	0

Table 5: Percentage of all events in the scenarios that were inferred to have occurred in "making tea", as the percentage of the observed low-level events vary.

Event Recognition

Low-level events, such as the ones shown in the last column of Table 4, are generated by fluents extracted from the vison subsystem. We evaluated the performance of our system using making tea sessions conducted by five different people from our annotated corpus. Over the five sessions, each session generated an average of 65 fluents and 18 low-level events.

The low-level visual subsystem detects location of objects in the scene (kettle, cup, and teabox), along with the location of the subject and hands in 3D space. The locations are quantifed into scene-independent visual fluents, which serve as triggers that generate low-level events. Table 6 shows the performance of low-level event detection for five selected sessions. Approximately two-thirds of the events were detected on average. Some high error counts were due to the fact that participants were not limited to a particular method of carrying on an activity and thus conducted actions that the low-level detection was not able to either capture or detect accurately. However, despite having multiple people doing the same high level activity in different ways, we show that our set of visual fluents is sufficient in recognizing a person making tea. We believe that as we are able to learn more low-level events, these errors can be minimized.

We infer mid-level and top-level events using our plan recognition framework. Providing a structure of making tea through the "D" predicates, we evaluated how well the system was able to identify and match the low-level events into the high level plan of making tea. The plan recognition system was able to "fill-in" many of the missing events, while dismissing irrelevant events that were not part of making tea as being *unexplained* (noted as *Corrected Low-level events*, or CLs), resulting in a significant improvement in recognition. These results are shown in Table 7.

Session	TP	FP	FN	P	R	F1
S1	11	4	7	.73	.61	.66
S2	14	14	4	.50	.78	.61
S3	13	3	5	.81	.72	.76
S4	13	5	5	.72	.72	.72
S5	8	6	10	.57	.44	.50
Total	59	32	31	.64	.66	.65

Table 6: Performance of "making tea" for the low-level event detection system.

Session	TP	FP	FN	CL	P	R	F1
S1	17	2	1	2	.89	.94	.91
S2	16	4	2	10	.80	.89	.84
S3	17	2	1	1	.89	.94	.91
S4	16	3	2	2	.84	.89	.86
S5	17	2	1	4	.89	.94	.91
Total	83	13	7	19	.86	.92	.89

Table 7: Performance of "making tea" plan recognition for each session. CL represents the number of corrected low-level events by the plan recognition system.

Next Steps

This paper provides only an interim report on our implemented system. As discussed in the introduction, a primary motivation for our framework was the goal of integrating verbal descriptions of events with the results of visual processing. We are updating the TRIPS parser so that the predicate names in the logical forms it produces are consistent with the predicates used in our event ontology. When this is complete, we will be able to perform a complete evaluation of plan recognition and tracking from video and speech.

Event tracking, however, is only the first step in our larger project of creating a system that can learn new complex activities from demonstration. We will formalize activity learning as the task of extending the domains of the event definition predicates so as to reduce the overall cost (*i.e.*, increasing the probability) of the observed demonstration.

References

Allen, J.; Swift, M.; and de Beaumont, W. 2008. Deep semantic analysis of text. In *Proc. Semantics in Text Processing*, STEP '08, 343–354.

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

Ansótegui, C.; Bonet, M. L.; and Levy, J. 2013. Sat-based maxsat algorithms. *Artifical Intelligence* 196.

Blackman, S. 1986. *Multiple-target tracking with radar applications*. Artech House radar library. Artech House.

Brendel, W.; Fern, A.; and Todorovic, S. 2011. Probabilistic event logic for interval-based event recognition. In *24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, 3329–3336.

Bui, H. H. 2003. A general model for online probabilistic plan recognition. In *Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-2003)*.

Comaniciu, D.; Ramesh, V.; and Meer, P. 2003. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* 25(5):564–575.

Dalal, N., and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, 886–893. Washington, DC, USA: IEEE Computer Society.

Gupta, S., and Mooney, R. J. 2010. Using closed captions as supervision for video activity recognition. In *Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2010)*.

Kautz, H. 1991. A formal theory of plan recognition and its implementation. In Allen, J.; Kautz, H.; Pelavin, R.; and Tennenberg, J., eds., *Reasoning About Plans*. Morgan Kaufmann Publishers. 69–126.

Kembhavi, A.; Yeh, T.; and Davis, L. 2010. Why did the person cross the road (there)? scene understanding using probabilistic logic models and common sense reasoning. In *11th European Conference on Computer Vision (EECV 2010)*.

Lei, J.; Ren, X.; and Fox, D. 2012. Fine-grained kitchen activity recognition using rgb-d. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp '12)*, 208–211.

Matuszek, C.; FitzGerald, N.; Zettlemoyer, L. S.; Bo, L.; and Fox, D. 2012. A joint model of language and perception for grounded attribute learning. In *29th International Conference on Machine Learning (ICML 2012)*.

Moore, D., and Essa, I. 2001. Recognizing multitasked activities using stochastic context-free grammar. In *In Proceedings of AAAI Conference*.

Morariu, V. I., and Davis, L. S. 2011. Multi-agent event recognition in structured scenarios. In *24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*.

Natarajan, S.; Bui, H. H.; Tadepalli, P.; Kersting, K.; and Wong, W. 2008. Logical hierarchical hidden Markov models for modeling user activities. In *In Proc. of ILP-08*.

Niu, F.; Ré, C.; Doan, A.; and Shavlik, J. W. 2011. Tuffy: Scaling up statistical inference in markov logic networks using an rdbms. *Proceedings of the VLDB Endowment (PVLDB)* 4(6):373–384.

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Mach. Learn.* 62(1-2):107–136.

Schmidt, C. F.; Sridharan, N. S.; and Goodson, J. L. 1978. The plan recognition problem: An intersection of psychology and artificial intelligence. *Artifical Intelligence* 11(1-2).

Shi, Y.; Huang, Y.; Minnen, D.; Bobick, A.; and Essa, I. 2004. Propagation Networks for Recognition of Partially Ordered Sequential Action. In *Proceedings of IEEE CVPR04*.

Singla, P., and Mooney, R. J. 2011. Abductive Markov Logic for plan recognition. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2011)*.

Swift, M.; Ferguson, G.; Galescu, L.; Chu, Y.; Harman, C.; Jung, H.; Perera, I.; Song, Y.; Allen, J.; and Kautz, H. 2012. A multimodal corpus for integrated language and action. In *Proc. of the Int. Workshop on MultiModal Corpora for Machine Learning*.

Tran, S., and Davis, L. 2008. Visual event modeling and recognition using markov logic networks. In *10th European Conference on Computer Vision (EECV 2008)*.

Viola, P., and Jones, M. J. 2004. Robust real-time face detection. *Int. J. Comput. Vision* 57(2):137–154.