Learning and Parsing Video Events with Goal and Intent Prediction

Mingtao Pei, Zhangzhang Si, Benjamin Yao, Song-Chun Zhu

Department of Statistics, University of California, Los Angeles.

Abstract

In this paper, we present a framework for parsing video events with stochastic Temporal And-Or Graph (T-AOG) and unsupervised learning of the T-AOG from video. This T-AOG represents a stochastic event grammar. The alphabet of the T-AOG consists of a set of *grounded spatial relations* including the poses of agents and their interactions with objects in the scene. The terminal nodes of the T-AOG are *atomic actions* which are specified by a number of grounded relations over image frames. An And-node represents a sequence of actions. An Or-node represents a number of alternative ways of such concatenations. The And-Or nodes in the T-AOG can generate a set of valid temporal configurations of atomic actions, which can be equivalently represented as a stochastic context-free grammar (SCFG). For each And-node we model the temporal relations of its children nodes to distinguish events with similar structures but different temporal patterns and interpolate missing portions of events. This makes the T-AOG grammar context-sensitive. We propose an unsupervised learning algorithm to learn the atomic actions, the temporal relations and the And-Or nodes under the *information projection principle* in a coherent probabilistic framework. We also propose an event parsing algorithm based on the T-AOG which can understand events, infer the goal of agents, and predict their plausible intended actions. In comparison with existing methods, our paper makes the following contributions. i) We represent events by a T-AOG with hierarchical compositions of events and the temporal relations between the sub-events. ii) We learn the grammar, including atomic actions and temporal relations, automatically from the video data without manual supervision. iii) Our algorithm infers the goal of agents and predicts their intents by a top-down process, handles events insertion and multi-agent events, keeps all possible interpretations of the video to preserve the ambiguities, and achieves the globally optimal parsing solution in a Bayesian framework; iv) The algorithm uses event context to improve the detection of atomic actions, segment and recognize objects in the scene. Extensive experiments, including indoor and out door scenes, single and multiple agents events, are conducted to validate the effectiveness of the proposed approach.

Keywords: Temporal And-Or Graph (T-AOG), Event Parsing, Unsupervised learning, Goal prediction, Information projection.

1. Introduction

1.1. Motivation and Objective

Cognitive studies [1] show that humans have a strong inclination to interpret observed behaviors of others as goal-directed actions. In this paper, we take such a teleological stance for understanding events in surveillance video, in which people are assumed to be rational agents [2] whose actions are planned to achieve certain goals. In this way, we infer the underlying goals and predict the next actions on the fly as the events unfold.

Imagine an office scene, where an agent picks up a cup, and walks to a desk on which there is a tea box. One might infer that his goal is to make a cup of tea, and one predicts that his next action is to put a tea bag in the cup. But instead, he picks up the phone on the desk, one then infers that his goal has been interrupted by an incoming call. After the call, he walks to a dispenser, and his action is obscured due to our viewing angle. After some time, he is observed drinking. One can now infer that he had poured water in the cup in the occluded time interval.

Daily videos contain a large variety of actions and events, which are defined through gestures and interactions between agents and environments. These action and event concepts constitute a large portion of human visual knowledge, therefore learning from video data is a promising way to acquire rich common sense knowledge.

To achieve the above event understanding capability, we need to address the following problems:

i) Events are compositional. An event can often consist of a sequence of actions and can be executed in multiple ways. Therefore a good representation must be hierarchical and account for temporal relations between sub-events.

ii) An inference algorithm must deal with event insertions, interruptions, multi-agent events and agent-object interactions. The inference process must also preserve the ambiguities both in the lower level atomic action detection and higher level event recognition to achieve globally optimized solution.

iii) A learning algorithm must discover the structure of the events from video data with minimal user supervision.

1.2. Overview of our work

In this paper, we represent events by Temperal And-Or Graph (T-AOG). The AOG was first introduced to compute vision in [3] and [4] for modeling visual objects, and has been used in [5] to analyze sports videos.

The T-AOG consists of a set of terminal nodes and And, Or-nodes. A terminal node specifies a contextual atomic action defined by a set of spatial relations (e.g. agent poses, agent's interaction with objects in the scene) grounded in the images. The And-nodes and Or-nodes represent verb concepts and are composed by the atomic actions. And-nodes represent temporal compositions of their children nodes. Or-nodes represent alternative ways to realize events, where each alternative has an associated probability to account for its branching frequency. With recursively defined And, Or-nodes, the T-AOG specifies a stochastic context free grammar (SCFG) whose language is the set of valid configurations of events. For each And-node, we model the temporal relations of its children nodes to distinguish events with similar structures but different temporal patterns and interpolate missing portions of events. This makes the T-AOG grammar context-sensitive.

We propose an inference algorithm for T-AOG based on the Earley Parser [6]. It finds the most likely parse graph by iterative bottom-up detection and top-down inference similar to the image parsing algorithm in [7]. Our inference algorithm is designed to have the capacity of handling interleaving events (e.g. event A interrupts event B) and online prediction of future events. Due to ambiguity arising from bottom-up detections, the parsing algorithm needs to keep a large number of parse graphs. For computational efficiency we prune the parse graphs at the time points corresponding to "deciding moments", so it is much more affordable than its counterpart in image grammar.

We propose an unsupervised learning algorithm to learn a T-AOG from video. The learning algorithm uses a recursive block pursuit procedure to generate terminal nodes and And-nodes from the data matrix of detected spatial relations. The ambiguity of bottom-up compositions is resolved during the recursive block pursuit. Then a graph compression procedure is then used to generate Or-nodes of T-AOG. The learning algorithm is guided by the information projection principle that minimizes the total description length.

1.3. Related work

Existing methods for event representation and recognition can be divided into two categories.

1) HMMs and DBN based methods. Brand et al. [8] modeled human actions by coupled HMMs. Natarajan [9] described an approach based on Coupled Hidden Semi Markov Models for recognizing human activities. Kazuhiro et. al. [10] built a conversation model based on dynamic Bayesian network. Al-Hames and Rigoll [11] presented a multi-modal mixed-state dynamic Bayesian network for meeting event classification. Although HMMs and DBN based algorithms achieved some success, the HMMs do not model the high order relations between sub-events, and the fixed structure of DBN limits its power of representation.
2) Grammar based methods. Ryoo and Aggarwal [12] used the context free grammar (CFG) to model and recognize composite human activities. Ivanov and Bobick [13] proposed a hierarchical approach using a stochastic context free grammar (SCFG). Joo and Chellappa [14] used

probabilistic attribute grammars to recognize multi-agent activities in surveillance settings. Zhang et al [15] applied an extended grammar approach to modeling and recognizing complex traffic events. These methods focus on the hierarchical structure of events, but the temporal relations between sub-events are not fully utilized. There are other methods for event representation and reasoning in the higher level, such as VEML and VERL [16, 17], and PADS [18].

In contrast to HMMs and DBN, the T-AOG can model higher order constraints than HMMs, while the Or-nodes enable the reconfiguration of the structures. So the T-AOG is more expressive than the fixed-structured DBN. The T-AOG also represents the temporal relations between multiple sub-events by the horizontal links between the nodes, so the resulting grammar is context-sensitive.

Most of the existing work predefine the event models manually and learn (or define) the parameters of the models for a predefined set of event classes. In contrast, we study an unsupervised learning algorithm that can generate richer event classes, reduce tedious manual labeling, thus provide more scalability for knowledge acquisition systems. Our work is inspired by recent progress in unsupervised learning and data mining [19, 20] as well as grammatical learning and inference [13, 21, 15] on video data. For event grammar learning, our strategy is most similar to Zhang et al. [15], which learns a stochastic context free grammar for trajectory analysis of multiple agents (e.g. vehicles in street intersections). In contrast, we adopt a richer feature representation including interactions between agents and environments. In addition, we append a Markov model of time constraints for adjacent events, resulting in a stochastic context sensitive grammar, which was introduced into computer vision by Zhu and Mumford in [4]. The stochastic T-AOG provides an efficient representation for knowledge extracted from video.

1.4. Main contributions

The contributions of our paper are:

1) We represent events by a T-AOG which represents the hierarchical compositions of events and the temporal relations between the sub-events.

2) We propose an unsupervised learning algorithm to learn the T-AOG automatically from video, based on the information projection principle.

3) Our parsing algorithm can afford to generate all possible parse graphs of single events, combine the parse graphs to obtain the interpretation of the input video, and achieve the global maximum-a-posteriori inference.

4) The agent's goal and intent at each time point is inferred by a bottom-up and top-down process based on the top-ranked parse graphs as the most probable interpretations. We show in human experiments that our parsing algorithm can correctly infer agent's goals and intents according to the video content.

5) We show that event context can be used to improve the detection result of atomic actions, and to better segment and recognize objects in the scene. We put the event learning and inference in the perspective of scene context, where there is a rich collection of agent-environment interactions. By inference on the joint probability of agent and environment events, we show how to use recognition of actions to help object recognition and scene segmentation.

6) We collect a video data set, which includes videos of daily life captured both in indoor and outdoor scenes to evaluate the proposed algorithm. The events in the videos include single-agent events, multi-agent events, and concurrent events. The results of the algorithm are evaluated by human subjects and our experiments show satisfactory results.

This paper is an enhanced combination of our previous conference papers [22] and [23] which focus on event parsing and grammar learning respectively. Here we integrate them into a coherent framework. We add more experimental results to evaluate the proposed algorithm, and new experiments on segmenting and recognizing objects in scene are shown in this journal paper.

2. Event representation by T-AOG

In this section, we introduce the T-AOG for event representation.

T-AOG is based on interactions between agents and objects in the scene. In the videos that we

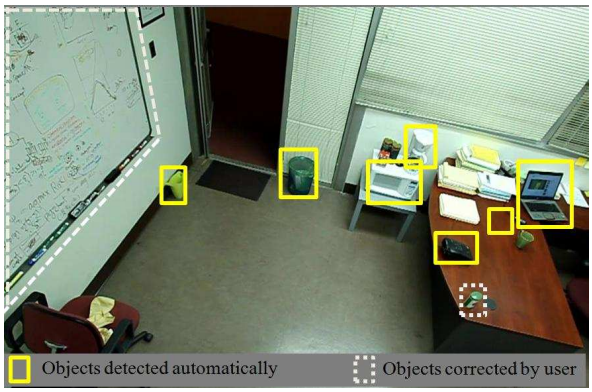

Figure 1: The detection result of objects in the office scene

collected, there are 13 classes of interest objects including mug, laptop, water dispenser in our training and testing data. These objects should be detected automatically, however, detection of multi-class objects in a complex scene cannot be solved perfectly by the state-of-art. Therefore, we adopt a semi-automatic object detection system. The objects in each scene are detected by the Multi-class boosting with feature sharing [24], and segmented by a recent indoor scene parsing algorithm [25]. This is not time consuming as it is done only once for each scene, and the objects of interest are tracked automatically during the video events. Figure 1 shows the detection result of the objects of interest in an office.

2.1. Grounded relations — the alphabet

The T-AOG is defined on a set of unary and binary relations which can be directly detected from video. We call these relations the *grounded relations*.

- A unary relation $r(A)$ is a time varying property of the agent or object A in the scene. As Figure 2 shows, it could be agent poses, e.g. Stand(person1) and Bend(person2), and object states, e.g. Open(door) and Closed(door).
- A binary relation $r(A, B)$ is the spatial relation (e.g. Touch(person1.hand, phone)) between A, B which could be agents, body parts (hands, feet), and objects. Figure 3 illustrates some typical relations.

In our experiments we use video data from relatively simple scenes with few people appearing at the same time. In this case, we can detect the spatial relations with minor ambiguity. It is beyond the scope of this paper to study complex behaviors in crowds (e.g. [26]).

Table 1 specifies the 24 unary and binary relations in the office scene. There are four types of relations: agent location ($r_{01} \sim r_{13}$), agent-environment interaction ($r_{14} \sim r_{17}$), agent pose ($r_{18} \sim r_{21}$) and environment event ($r_{22} \sim r_{24}$). Here we do not use the "Off" relation as shown in Figure 3 since we can infer the status of "Off" from the status of "On". The details of how these relations are detected are explained in Section 3.2.

Status of person	Symbols	Examples	Status of objects	Examples
Stand(P1)			On (phone)	
Stretch(P1)			Off (phone)	
Bend (P1)			On(screen)	
Sit (P2)			Off(screen)	

Figure 2: Some unary relations. The left part of the table shows the four unary relations as agent poses, including 'Stand', 'Stretch', 'Bend' and 'Sit'. The right part shows the two fluents ('On' and 'Off') of the phone and the screen of laptop.

Binary Fluent (A,B)	Touch (A,B)	Near (A,B)	Occlude (A,B)	In(A,B)
Symbols				
Examples				

Figure 3: Some binary relations between agents (parts) and background objects.

2.2. Atomic actions — the terminal nodes

An atomic action is a vector of grounded relations $a = (r_1, ...r_J)$ that happen sequentially in the joint domain of space and time.

Figure 4 shows three atomic actions defined on the grounded relations. Table 2 shows the atomic

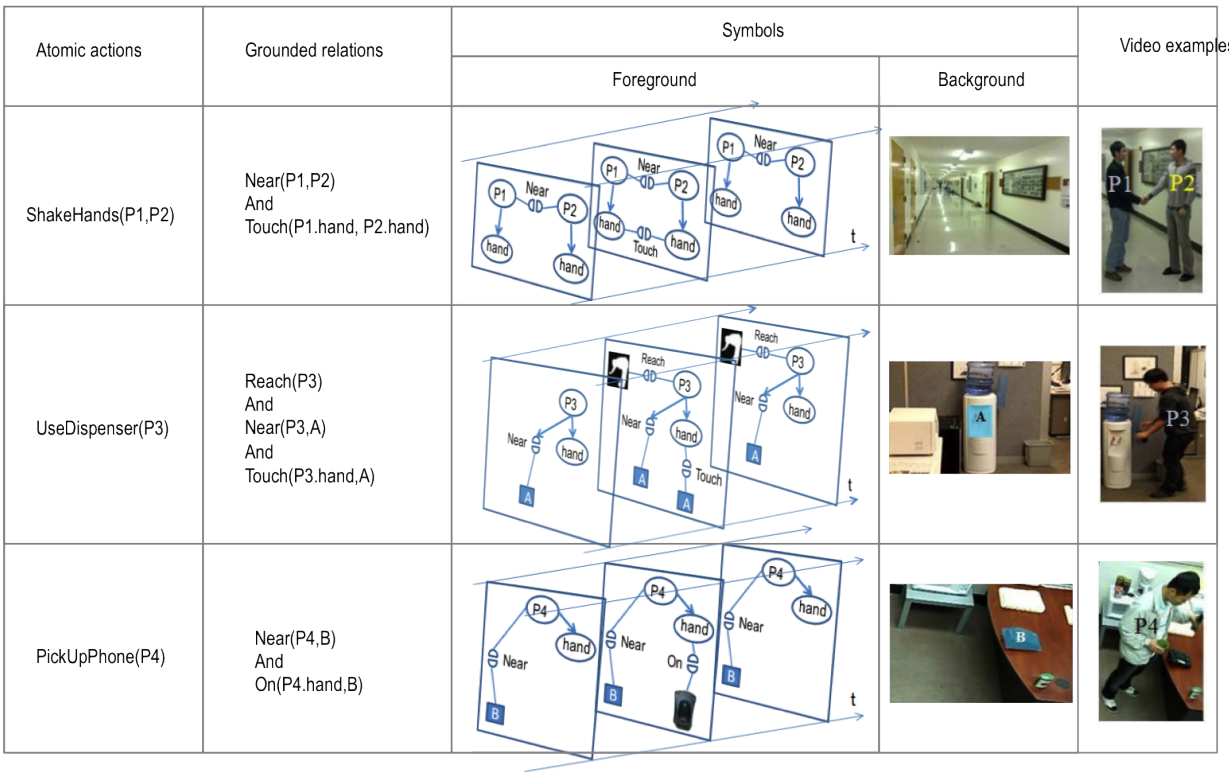

Figure 4: Some atomic actions. Each atomic action is defined on a set of grounded relations shown by 2 half circles. Unary relations 'Bend' and 'On' are defined in Figure 2. Binary relations 'Near' and 'Touch' are defined in Figure 3. For the atomic action 'ShakeHands', when P1 is considered as the agent, P2 is regarded as object and vice versa. See [27] for a more sophisticated system to detect agent poses and interactions with the scene.

actions used in the office scene. These atomic actions are learned automatically from the training data. The learning process is explained in Section 3.

An atomic action is detected when all its relations are detected with probability higher than a given threshold, and the probability of the atomic action is computed as the product of the probabilities of all its constituent relations. An atomic action $a = (r_1, ..., r_J)$, has the following probability given a short video snipet $I_{1:t}$,

$$p(a \mid I_{1:t}) = \frac{1}{Z} \prod_{j=1}^{J} p(r_j) \propto \exp\{-E(a)\} \quad (1)$$

where

$$E(a) = -\sum_{j=1}^{J} \log p(r_j)$$

is the energy of a and Z is the normalizing constant for all atomic actions. We use $n = 26$ learned atomic actions shown in Table 2.

In our experiments, we only detect several simple agent poses (e.g. standing, sitting) as we focus on interactions between agents and objects in the scene. In future work, we will extend our experiments to detect a richer collection of more sophisticated agent poses using animated AND-OR Templates [27].

Given the input video $I_\wedge$ in a time interval $\wedge = [0, T]$, multiple atomic actions are detected with probabilities to account for the ambiguities in the grounded relations contained in the atomic actions, for example, the relation 'Touch(A,B)' cannot be clearly differentiated from the relation 'Near(A,B)' unless kinect data is used. The other reason is the inaccuracy of foreground detection. Fortunately, most of the ambiguities can be removed by the event context in the top-down bottom-up inference, we will show this in the experiment section.

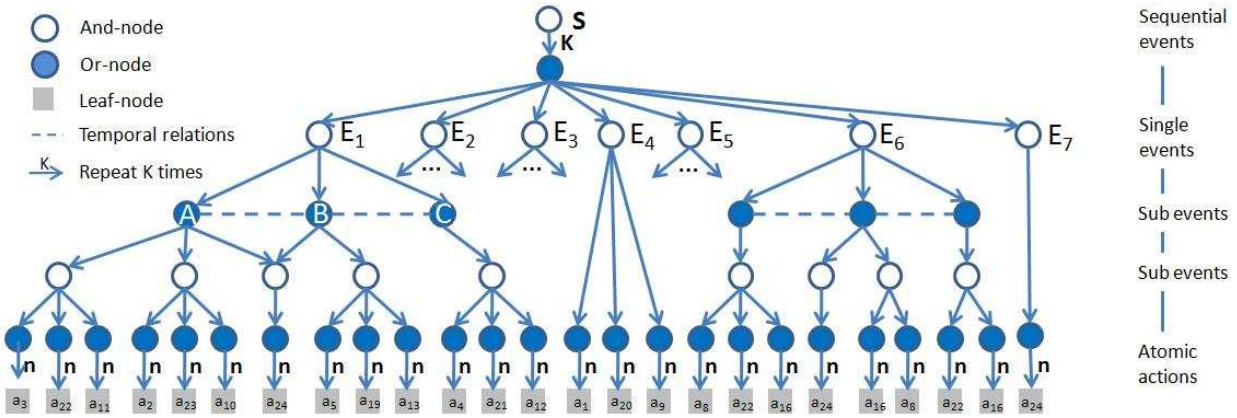

Figure 5: T-AOG for events in the office scene. S is the root node which represents the sequential events happened in the office. It is a Set-node and could be any combinations of K single events. For example, S could be $E_1|E_2|E_1E_2|E_3E_2E_3|....$ $E_1, ..., E_7$ are And-nodes representing single events. The atomic actions are also represented by Set-nodes, and could last for 1 to n frames. The temporal relations are given by the ratio of the lasting time between related nodes. For clarity, only the temporal relations between sub-events are shown.

2.3. The T-AOG for events

An T-AOG (see Figure 5 for an example) is specified by a 6-tuple

$$\text{T} - \text{AOG} =< S, V_N, V_T, R, \Sigma, P > .$$

S is the root node for an event category, $V_N = V^{and} \cup V^{or}$ is the set of non-terminal nodes (events and sub-events) composed of an And-node set and an Or-node set.

Each And-node represents an event or sub-event, and is decomposed into sub-events or atomic actions as their children nodes. The children nodes must occur in a certain temporal order.

An Or-node has a number of alternative ways to realize an event or sub-event, and each alternative has a probability associated with it to indicate the frequency of occurrence. A Set-node is a special Or-node which can repeat m times with probability $p(m)$ and accounts for the time warping effects.

V_T is a set of terminal nodes for atomic actions. R is a number of relations between the nodes (temporal relations), Σ is the set of all valid configurations (possible realizations of the events) derivable from the T-AOG, i.e. its language, and P is the probability model defined on the graph. The T-AOG for events in the office scene is shown in Figure 5. These events are learned from the training data automatically which is illustrated in the next section.

2.4. Non-parametric temporal relations

The And-nodes have already defined the temporal order of its children-nodes, and the Set-nodes representing atomic actions have modeled the lasting time of the atomic action by the frequency of its production rules. Here we augment the T-AOG by adding temporal constraints between related nodes.

Unlike [13] and [15] which use Allen's 7 binary temporal relations [28], we use non-parametric filters to model the relations between multiple nodes. We use the T-AOG of E_1 shown in Figure 5 to illustrate the temporal relations. E_1 is an And-node and A, B and C are three sub-nodes; τ_A, τ_B and τ_C are the lasting time of A, B and C, respectively. There is a constraint between the lasting time of A, B and C. For example, when an agent does event E_1 in a hurry, the lasting time of A, B and C will be shorter than usual, while the ratio of the lasting time between A, B and C will remain stable. This relation r is modeled by a distribution of a function response over the nodes included in the relation. We use $\tau_{E_1} = (\tau_A, \tau_B, \tau_C)$ to represent the lasting time of E_1, and $F = (F_1, F_2, F_3)$ to represent the function on which the response of τ_{E_1} is modeled, F could be regarded as a filter and $< \tau_{E_1}, F >$ could be regarded as a filter response. We use histogram to model the distribution of the response, and the F^*, on which the distribution of the training data's response has the minimum entropy, are selected to model the relation as in [4]. Given τ and F^*, the probability of the relation r is

Table 1: The grounded spatial relations of T-AOG: directly detectable from video.

Name	Definition	Description
r_{01}	absent(agent)	not found in the frame
r_{02}	near(agent, other_agent)	near other agent
r_{03}	near(agent, board)	near the white board
r_{04}	near(agent, door)	near the door
r_{05}	near(agent, dispenser)	near the water dispenser
r_{06}	near(agent, trash_can)	near the trash can
r_{07}	near(agent, mug)	near the mug
r_{08}	near(agent, laptop)	near the laptop
r_{09}	near(agent, phone)	near the phone
r_{10}	near(agent, basin)	near the basin
r_{11}	near(agent, microwave)	near the microwave
r_{12}	near(agent, tea_box)	near the tea box
r_{13}	in(agent, door)	agent is in the door
r_{14}	touch(agent, keyboard)	typing on keyboard
r_{15}	touch(agent, mug)	grabbing the mug
r_{16}	touch(agent, phone)	grabbing the phone
r_{17}	touch(agent, tea_box)	grabbing the tea box
r_{18}	bend(agent)	bend down
r_{19}	sit(agent)	sitting on something
r_{20}	raise_arm(agent)	raising arm
r_{21}	stand(agent)	standing straight
r_{22}	occlude(soccer match, screen)	soccer match on the screen
r_{23}	on(phone)	phone is in use
r_{24}	on(screen)	screen is on

Table 2: Learned atomic actions.

Node Name	Semantic Name	Contained relations
a_{01}	absent	r_{01}
a_{02}	arrive at door	r_{04}, r_{21}
a_{03}	enter door	r_{04}, r_{21}, r_{13}
a_{04}	stand near phone	r_{09}, r_{21}
a_{05}	sit near phone	r_{09}, r_{19}
a_{06}	stand and use phone	$r_{09}, r_{21}, r_{16}, r_{23}$
a_{07}	sit and use phone	$r_{09}, r_{19}, r_{16}, r_{23}$
a_{08}	arrive at trashcan	r_{06}, r_{21}
a_{09}	throw trash	r_{06}, r_{18}
a_{10}	arrive at basin	r_{10}, r_{21}
a_{11}	dump water	r_{10}, r_{18}, r_{15}
a_{12}	arrive at dispenser	r_{05}, r_{21}, r_{15}
a_{13}	use dispenser	r_{05}, r_{18}, r_{15}
a_{14}	arrive at tea box	r_{12}, r_{21}, r_{15}
a_{15}	use tea box	$r_{12}, r_{21}, r_{15}, r_{17}$
a_{16}	arrive at board	r_{03}, r_{21}
a_{17}	discussion	r_{03}, r_{21}, r_{02}
a_{18}	arrive at laptop	r_{08}, r_{21}
a_{19}	sit near laptop	r_{08}, r_{19}
a_{20}	watch soccer	$r_{08}, r_{19}, r_{22}, r_{24}$
a_{21}	celebrate	$r_{08}, r_{20}, r_{22}, r_{24}$
a_{22}	use laptop	$r_{08}, r_{19}, r_{14}, r_{24}$
a_{23}	arrive at microwave	r_{11}, r_{21}
a_{24}	use microwave	r_{11}, r_{18}
a_{25}	arrive at mug	r_{07}, r_{19}
a_{26}	take mug	r_{07}, r_{19}, r_{15}

$$p(r) \sim h(< \tau, F^* >) \quad (2)$$

where h is the histogram of the training data' s response over F^*. One may use multiple F to model the relations if needed.

2.5. Parse graph

A parse graph is an instance of the T-AOG obtained by selecting variables at the Or-nodes and specifying the attributes of And-nodes and terminal nodes. We use pg to denote the parse graph of the T-AOG of a single event E_i. We denote the following components in pg:

- $V^t(pg) = \{a_1, ..., a_{n_t(pg)}\}$ is the set of leaf nodes in pg.
- $V^{or}(pg) = \{v_1, ..., v_{n_{or}(pg)}\}$ is the set of non-empty Or-nodes in pg, $p(v_i)$ is the probability that v_i chooses its sub-nodes in pg.
- $R(pg) = \{r_1, ..., r_{n(R)}\}$ is the set of temporal relations between the nodes in pg. Without temporal relations, the pg reduces to a parse tree.

The energy of pg is defined as in[4]:

$$\begin{aligned} \varepsilon(pg) \quad = \quad & \sum_{a_i \in V^t(pg)} E(a_i) + \sum_{v_i \in V^{or}(pg)} -\log p(v_i) \\ & + \sum_{r_i \in R(pg)} -\log p(r_i) \end{aligned} \quad (3)$$

The first term is the data term. It expresses the energy of the detected terminal nodes (atomic actions) which is computed by Eq. 1. The second term is the frequency term. It accounts for how frequently each Or-node decomposes in a certain way, and can be learned from the training data. The third term is the relation term which models the temporal relations between the nodes in pg and can be computed by Eq. 2.

Given input video $I_\wedge$ in a time interval $\wedge = [0, T]$. We use PG to denote parse graph for a sequence of events in S and to explain the $I_\wedge$. PG is of the following form,

$$PG = (K, pg_1, ..., pg_K)$$

where K is the number of parse graphs for events.

3. Learning the T-AOG

3.1. Information projection

The unsupervised learning of stochastic T-AOG is conducted under the information projection and minimum description length principle [23]. Here we provide a review of the related theoretical instruments.

Let $\mathcal{X}_+ = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ be positive examples (e.g. observed video clips) governed by an unknown target distribution $f(\mathbf{x})$. Let $\mathcal{X}_-$ be a large set of random negative examples governed by a reference distribution $q(\mathbf{x})$ (here q is an i.i.d. uniform distribution). For each example $\mathbf{x}$, a list of spatial relations

$$(r_1(\mathbf{x}), r_2(\mathbf{x}), ..., r_D(\mathbf{x}))$$

are extracted from the video clip. These relations form a predefined alphabet, just like the set of weak classifiers in adaboost. Our objective of learning is to pursue a model $p(\mathbf{x})$ to approximate $f(\mathbf{x})$ in a series of steps:

$$q(\mathbf{x}) = p_0(\mathbf{x}) \to p_1(\mathbf{x}) \to \cdots p_T(\mathbf{x}) = p(\mathbf{x}) \approx f(\mathbf{x})$$

starting from q.

The above model updates are performed by selecting a most informative subset from all the spatial relations. The model p after T iterations contains T selected spatial relations $\{r_t : t = 1, ..., T\}$. If the selected spatial relations capture all the related information about the scene semantics in $\mathbf{x}$, it can be shown by variable transformation [29] that:

$$\frac{p(\mathbf{x})}{q(\mathbf{x})} = \frac{p(r_1, ..., r_T)}{q(r_1, ..., r_T)}.$$

So p can be constructed by reweighting q with the marginal likelihood ratio on selected spatial relations.

Under the maximum entropy principle, $p(\mathbf{x})$ can be expressed in the following log-linear form:

$$p(\mathbf{x}) = q(\mathbf{x}) \prod_{t=1}^{T} \left[\frac{1}{z_t} \exp\left\{\beta_t r_t(\mathbf{x})\right\} \right]. \quad (4)$$

where β_t is the parameter for the t-th selected spatial relation r_t and z_t ($z_t > 0$) is the individual normalization constant determined by β_t:

$$z_t = \sum_{r_t} q(r_t) \exp\{\beta_t r_t\}.$$

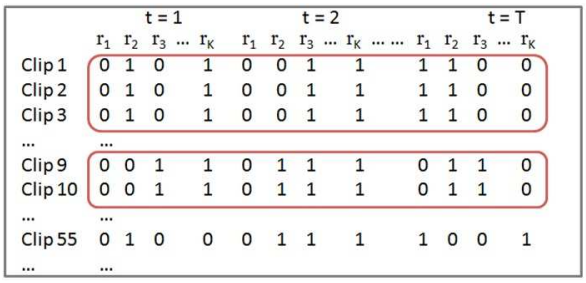

Figure 6: Pursuing homogeneous blocks from the data matrix. Each block corresponds to a terminal node or an And-node in T-AOG.

By the information projection principle [30, 31, 29], we adopt a step-wise procedure to for selecting spatial relations. In particular, the t-th spatial relation r_t is selected and model p_t is updated by:

$$\begin{aligned} p_t &= \arg\min \quad \mathcal{K}(p_t|p_{t-1}) \\ s.t. \qquad & E_{p_t}[r_t] = \frac{1}{N}\sum_{i=1}^{N} r_t(\mathbf{x}_i) \end{aligned} \tag{5}$$

where $\mathcal{K}$ denotes the Kullback-Leibler divergence, and by minimizing it we select a most informative spatial relation r_t to augment p_{t-1} towards p_t. The constraint equation in Eq. (5) ensures that the updated model is consistent with the observed training examples on marginal statistics. The optimal β_t can be found by a simple line search or gradient descent to satisfy the constraint in Eq. (5).

3.2. Block pursuit on data matrix

Data matrix. Firstly we set up a data matrix $\mathbf{R}$ using spatial relations of positive training examples as shown in Figure 6. Each row of $\mathbf{R}$ is the vector of spatial relations detected from one example (or video clip) in $\mathcal{X}_+$. For simplicity, we assume all positive training examples are aligned and have the same dimensionality. Therefore $\mathbf{R}$ is a matrix with N (number of positive examples) rows and D (number of all candidate spatial relations) columns, and each entry

$$\mathbf{R}_{ij} = r_j(\mathbf{x}_i)$$

is a binary response. $\mathbf{R}_{ij} = 1$ means the spatial relation j holds in example $\mathbf{x}_i$.

Block pursuit. On the data matrix, we pursue large homogeneous blocks $\{\mathcal{B}_k : k = 1, ..., K\}$. A block is specified by a set of common spatial relations (columns) that co-occur in a set of examples (rows). Each block corresponds to a frequent verb concept, i.e. an terminal node or And-node composed by several spatial relations. For example, the verb concept a_{02} (arrive at the door) in Table 2 is composed by two spatial relations: near(agent, door) and stand(agent). The verb concept emerges from data because it appears frequently and with high confidence, thus it is readily represented by an AND node that strongly binds its constituent relations. Quantitatively, we can measure this by the information gain of block $\mathcal{B}_k$, computed by the summation over the block:

$$\text{Gain}(\mathcal{B}_k) = \sum_{\substack{i \in \text{rows}(\mathcal{B}_k) \\ j \in \text{cols}(\mathcal{B}_k)}} (\beta_{k,j}\mathbf{R}_{i,j} - \log z_{k,j}) \tag{6}$$

where rows$(\cdot)$ and cols$(\cdot)$ denote the rows and columns of block $\mathcal{B}_k$. cols$(\mathcal{B}_k)$ correspond to the selected spatial relations, capturing their co-occurrence in space and time. And rows$(\mathcal{B}_k)$ are the examples that belong to the k-th block. $\boldsymbol{\beta}_{k,j}$ is the multiplicative parameter of selected spatial relation j, and $z_{k,j}$ is the individual normalizing constant determined by $\boldsymbol{\beta}_{k,j}$. Eq. (6) measures the information gain by explaining the submatrix covered by $\mathcal{B}_k$ using the foreground model p instead of the background model q. Similar approaches have also been adopted in the grammar learning of textual data [32].

Recall that we pursue a series of models starting from $q(\mathbf{x})$ to approximate the target distribution $f(\mathbf{x})$ governing training positives $\mathcal{X}_+$. This corresponds to maximizing the log-likelihood $\log p(\mathbf{x})$ on $\mathcal{X}_+$. Initially $p = q$, and the data matrix has a log-likelihood $L_0(\mathbf{R})$. After pursuing K blocks, the resulting image log-likelihood is:

$$L = L_0 + \sum_{k=1}^{K} \text{Gain}(\mathcal{B}_k). \tag{7}$$

The block pursuit algorithm is a greedy procedure that maximizes the log-likelihood in Eq. (7). Each time we permute rows and columns the data matrix to pursue the block with the largest gain as computed in Eq. (6). The entries covered by the block are then explained away and excluded from subsequent block pursuit. This procedure is repeated until the information gain of the newly pursued block is negligible.

To penalize the model complexity, we apply a constant penalty for each additional block learned.

This is equivalent to imposing a Laplacian prior on the size of the learned grammar.

The above block pursuit procedure can be implemented either by clustering, which produces multiple blocks or non-terminal nodes at the same time, or by stepwise pursuit, which produces one block or non-terminal node at a time.

The block pursuit procedure for T-AOG is carried out into two stages. (1) Learn a set of terminal nodes as blocks on the data matrix of grounded spatial relations. These terminal nodes account for *atomic events* which directly specify spatial temporal configurations of grounded relations. This is done by clustering. (2) Learn non-terminal nodes as blocks on the data matrix of atomic actions, to account for longer events composed of atomic actions.

3.3. Detecting grounded spatial relations

As a preprocessing step, we perform one round of bottom up detection for grounded spatial relations.

Firstly we use a standard background subtraction algorithm to segment moving agent and fluent changes of objects, and use a commercial surveillance system to track the detected agent.

The relations of agents' location ($r_0 \sim r_{13}$) are detected by the distance between agent and objects which belongs to normal distribution. The location of the agent is detected by combining foreground segmentation and skin color detection that locates head and hands of the agent . Then the distance between agent and objects is computed directly as the location of objects are known (automatically detected or manually labeled).

The agent pose is inferred by a nearest neighbor classifier using both pixels and foreground segmentation map within the estimated bounding box for the agent. An illustration of four poses using segmented foreground mask is shown in Figure 7. The agent-environment interaction `touch(agent, keyboard)` and `touch(agent, phone)` are detected by checking whether there is enough skin color within the designated area for the laptop and phone, which are static objects in the office environment. The relation `touch(agent, mug)` and `touch(agent, tea box)` are also detected using skin color, and also the unique color and shape of the mug and tea box. When a relation involves an object, the object is tracked until the relation finishes and the new position of the object will be updated.

The environment relations `occlude(soccer match, screen)` is determined by checking whether there is large amount of green color within the designated area of laptop. The `on` relations are detected by the properties of the object area such as intensity histogram of the bounding box.

Using the techniques described above, we detect grounded relations for every video frame. The detection result is organized as a spatial temporal table where each row corresponds to a time frame. Each column corresponds to a grounded relation.

Figure 7: Standing, bending, sitting and raising-arm poses.

3.4. Learning atomic actions

We define *atomic actions* to be simple and transient events composed spatially and temporally by grounded relations. To learn an alphabet of atomic actions, we use a temporal scanning window spanning 5 frames to collect a large number of small clips. Each 5-frame clip is described by the detected relation vector:

$$\{(\mathbf{r}_{1,1}, ..., \mathbf{r}_{1,D}, ..., \mathbf{r}_{5,1}, ..., \mathbf{r}_{5,D})\}$$

where $D = 24$ is the number of grounded relations detected per frame. A k-means clustering is then performed on the grounded relation vectors of these 5-frame clips, using the simple Hamming distance as the metric. And a centroid of a cluster is simply determined as the grounded relation vector that has minimal distance to all the cluster members. As the times pan is very small, we can assume that the grounded relations (*e.g.* agent location, pose) stay constant during the short period. So we constrain the centroids to be stationery, *i.e.* $\mathbf{r}_{1,d} = \mathbf{r}_{2,d} = ...\mathbf{r}_{5,d}, \forall d = 1, ...D$. For each cluster, we estimate the symbol probabilities $p(\mathbf{r}_1)$, ..., $p(\mathbf{r}_{24})$ by counting the member sub-sequences of the cluster. And we represent this stochastic model by its mode (the most likely sub-sequence) as the cluster prototype $\mathbf{r}^{(k)}_{1:24}$ for brevity. Each cluster corresponds to a block pursued in the data matrix in Figure 6.

The result of clustering is a list of 26 atomic actions shown in Table 2. Each atomic action is represented by a list of grounded relations that are activated. The semantic description for these atomic

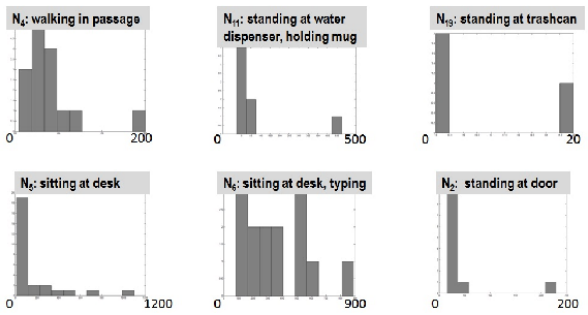

Figure 8: The duration model for the length of repetition.

actions is in Table 2. The atomic actions that happen most frequently include a_{19} (sit near laptop), a_{22} (use laptop), a_{20} (watch soccer) and a_{03} (enter door). a_{19}, a_{22} can be considered as constituent components of a longer event "working by laptop". a_{03} indicates the student is entering or leaving. The the learned atomic actions and their relative frequencies are representative and truthful to the video data.

Now the sequence of multi-dimensional relations is encoded by the alphabet of 26 atomic actions. For the computational efficiency in discovering longer events, we use hard assignments by computing the most likely atomic action per every 5 frames. The resulting sequence of atomic actions is

$$\mathbf{w}_{1:T} = (w_1, ..., w_T), \qquad \text{where } w_t \in \{a_{01}, ..., a_{26}\}$$

and T is the total number of video frames divided by 5.

3.5. Learning longer events and T-AOG

There is large variation in the duration of atomic actions. For example, a student may repeatedly enter the office, work for a varying time and leave the office. If we naively group atomic actions into longer ones, we get a large number of repetitive patterns of various lengths, providing little information. To deal with temporal variation, we perform a simple compression operation: every repetitive subsequence is summarized into one symbol (e.g. *bbbb* substituted by *b*). We may interpret this operation as learning a large number of grammar rules in the form $\tilde{N} \rightarrow NN...N$ with various lengths of repetition. We estimate a non-parametric model (Figure8) for the length of repetition, or *duration* under maximum likelihood principle.

After compression, the original sequence of atomic actions $\mathbf{w}_{1:T}$ is transformed into a much shorter one $\mathbf{c}_{1:M}$ ($M << T$) where each symbol c_i takes value from the same domain as w_i.

There will be some frames that non of the relations are activated except r_{21}, that is, in these frames the agent just stand somewhere that not near any interested objects. These frames are regarded as background frames, that is during these frames, no interest event or action happened. The background frames and the frames in which absent is detected are used to separate the video into different sequences, each sequence is an single event. One example sequence (event) is a_{01}, a_{02}, a_{03}, it is the entering event, which composed of absent, arrive at door and enter door. Another example is $a_{25}, a_{26}, a_{14}, a_{15}, a_{12}, a_{13}$. It is the taking water event, which composed of arrive mug, take mug, arrive at tea box, use tea box, arrive dispenser, use dispenser. These sequences are used to learn the grammar.

We then scan the sequence $\mathbf{c}_{1:M}$ to collect subsequences of length l ($l = 2$ in our system) and form a data matrix. Now the columns of this data matrix are atomic actions instead of grounded relations. A large number of homogeneous blocks (*i.e.* frequent sub-sequences) are identified from the data matrix. They are candidates for the right hand side of production rules in the event grammar. From the candidates, we select a subset of production rules in a step wise fashion.

The proposed candidate production rule takes the form $\alpha \rightarrow \beta\gamma$. It re-encodes the current sequence into a new sequence by replacing all occurrences of $\beta\gamma$ by α. By doing this, the reduction in description length is computed as:

$$\text{reduction} = \Delta_1 + \Delta_2 + \Delta_3 - \text{constant} \qquad (8)$$

and,

$$\Delta_1 = n'_\alpha \cdot \Big(\log \frac{n_\alpha}{n'}) - \log \frac{n_\beta}{n} - \log \frac{n_\gamma}{n}\Big)$$

$$\Delta_2 = n'_\beta \cdot \left(\log \frac{n'_\beta}{n'} - \log \frac{n_\beta}{n}\right) + n'_\gamma \cdot \left(\log \frac{n'_\gamma}{n'} - \log \frac{n_\gamma}{n}\right)$$

$$\Delta_3 = (n' - n'_\beta - n'_\gamma - n'_\alpha) \cdot \log \frac{n}{n'}$$

where $n'_\alpha, n'_\beta, n'_\gamma$ are the frequencies of α, β, γ in the new sequence respectively, n_β, n_γ are the corresponding frequencies in the current sequence. n is the length of the current sequence. $n' = n - n'_\alpha$ is the length of the new sequence. We rank the candidate production rules using Eq.8 and select the

Table 3: Learned production rules of T-AOG. For simplicity, we omit the starting symbol S and the branching probabilities that S produces the following non-terminal nodes.

Production rule	Semantic
$N_{01} \rightarrow a_{01}a_{02}a_{03}a_{02}$	absent, arrive at door, enterdoor, arrive at door
$N_{02} \rightarrow a_{02}a_{03}a_{02}a_{01}$	arrive at door, enter door, arrive at door, absent
$N_{03} \rightarrow a_{04}a_{06}$	stand near phone, stand and phone
$N_{04} \rightarrow a_{05}a_{07}$	sit near phone, sit and use phone
$N_{05} \rightarrow a_{25}a_{26}$	arrive at mug, take mug
$N_{06} \rightarrow a_{10}a_{11}$	arrive at basin, tdump water
$N_{07} \rightarrow a_{14}a_{15}$	arrive at tea box, use tea box
$N_{08} \rightarrow a_{12}a_{13}$	arrive at dispenser, use dispenser
$N_{09} \rightarrow a_{26}a_{25}$	take mug, arrive at mug
$N_{10} \rightarrow N_{05}N_{06}N_{07}N_{08}N_{09}$	take mug, dump water, make tea, take water, take mug
$N_{11} \rightarrow N_{05}N_{07}N_{08}N_{09}$	take mug, make tea, take water, take mug
$N_{12} \rightarrow N_{05}N_{08}N_{09}$	take mug, take water, take mug
$N_{13} \rightarrow N_{05}N_{06}N_{08}N_{09}$	take mug, dump water, take water, take mug
$N_{14} \rightarrow a_{18}a_{19}a_{22}$	arrive at laptop, sit near laptop, use laptop
$N_{15} \rightarrow N_{14}a_{19}$	use laptop, sit near laptop
$N_{16} \rightarrow a_{20}a_{21}$	watch soccer, celebrate
$N_{17} \rightarrow N_{14}N_{16}, a_{19}$	use laptop, watch soccer, sit near laptop

largest one. This learning procedure is recursively carried out, until the reduction of description length is too small for any new candidate production rule.

As a result, we obtain a dictionary of new production rules shown in Table 3, where to make the grammar more compact we merge shorter production rules into a longer ones that maximally reduce the description length.

We can see from the table that $N_{10} \sim N_{13}$ are taking water events, we can cluster them by the objects involved in them, the mug. Similarly, N_{15} and N_{17} are clustered by the laptop. Then, we can align them to learn the OR-Node. We introduce a special event (action) "NULL" to represent that the NULL event(action). It represents the event or action that is not interested. We put NULL event in the aligned sequence as show in Figure. 9, and by combining the production rules (*e.g.* $N_4 NULL \cup N_4 N_5 \rightarrow N_4(NULL \cup N_5)$) we get a stochastic T-AOG for each clustered event. The T-AOG of the take water event is illustrated in Figure. 10, where for brevity we only show the graph structure and omit the branching probabilities of OR nodes. Here an AND node represents an event is decomposed into sub-events or atomic actions; an OR node represents alternative ways to realize an event. The T-AOG presents a large amount of node sharing in the compositional hierarchy.

The terminal nodes $\{a_1, a_2, ...\}$ and non-terminal And-nodes form a compositional hierarchy. By learning them altogether, we greatly reduce the ambiguity of segmenting video into events and atomic actions.

3.6. Learning the parameters of T-AOG

After the structure (i.e. And-Or nodes) of T-AOG is learned, we can compute the probability of each branch of OR-Node by counting the time each branch appears. This is essentially a maximum likelihood estimation. The details can be found in [4]. Let V_i^{or} be an Or-node and v be an index of one of V_i^{or}'s branches, then

$$p(V_i^{or} = v) = \frac{\sum_{pg \in PG} \mathbf{1}_{V_i(pg)=v}}{|PG|}$$

where PG is the set of all parse graphs on the training data.

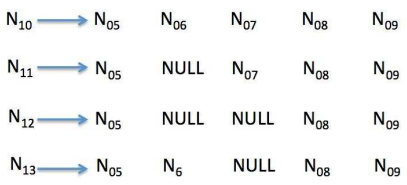

Figure 9: The aligned rules of fetching water.

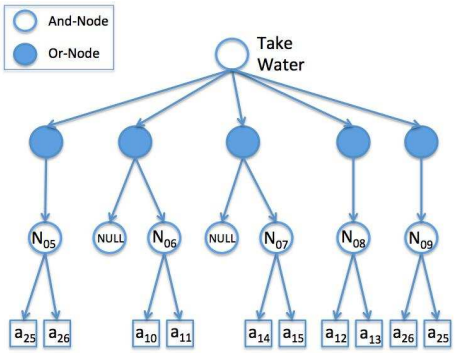

Figure 10: The learned T-AOG of fetching water.

4. Event parsing with Goal inference and Intent Prediction

In this section, we first show the event parsing process by assuming that there is only one agent in the scene in Section 4.1 - 4.3. In Section 4.4 we show how to parse events when there are multiple agents in the scene.

4.1. Formulation of event parsing

The input of our algorithm is a video $I_\wedge$ in a time interval $\wedge = [0, T]$, and atomic actions are detected at every frame I_t. We denote by $\wedge_{pg_i}$ the time explained by parse graph pg_i. $PG = (K, pg_1, ..., pg_K)$ is regarded as an interpretation of $I_\wedge$ where

$$\begin{cases} \cup_{i=1}^{K} \wedge_{pg_i} = \wedge \\ \wedge_{pg_i} \cap \wedge_{pg_j} = \emptyset \quad \forall ij \quad i \neq j \end{cases} \tag{9}$$

We use a small T-AOG in Figure 11(a) to illustrate the algorithm. Figure 11(b) shows a sample input of atomic actions. Note that there are multiple atomic actions at each time point. Figure 11(c), (d) and (e) show three possible parse graphs (interpretations) of the input up to time t_4. $PG_1 = (1, pg_1)$ in figure 11(c) is an interpretation of the video $I_{[t_1,t_4]}$ and it segments $I_{[t_1,t_4]}$ into one single event E_1 at the event level, and segments

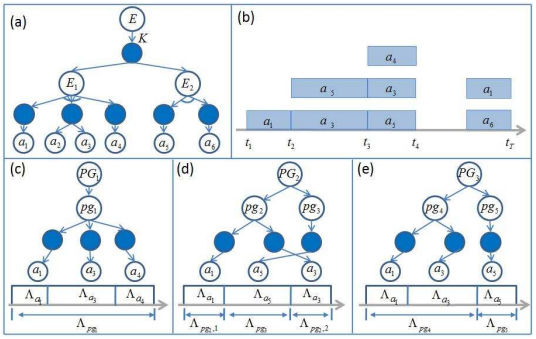

Figure 11: (a) A small T-AOG. (b) A typical input of the algorithm. (c),(d) and (e) are three possible parse graphs (interpretations) of the video $I_{\wedge[t_1,t_4]}$. Each interpretation segments the video $I_{\wedge[t_1,t_4]}$ into single events at the event level and into atomic actions at the atomic action level.

$I_{[t_1,t_4]}$ into three atomic actions a_1, a_3 and a_4 at the atomic action level. $PG_2 = (2, pg_2, pg_3)$ in Figure 11(d) segments $I_{[t_1,t_4]}$ into two single events E_1 and E_2, where E_2 is inserted in the process of E_1. Similarly $PG_3 = (2, pg_4, pg_5)$ in 11(e) is another parse graph and segments $I_{[t_1,t_4]}$ into two single events E_1 and E_2.

We can see that the segmentation of events is automatically integrated in the parse process and each interpretation could segment the video $I_\wedge$ into single events, and remove the ambiguities in the detection of atomic actions by the event context. The energy of PG is

$$E(PG \mid I_\wedge) = p(K) \sum_{k=1}^{K} (\varepsilon(pg_k \mid I_{\wedge_{pg_k}}) - \log p(k)) \tag{10}$$

where $p(k)$ is the prior probability of the single event whose parse graph in PG is pg_k, and $p(K)$ is a penalty item that follows the poisson distribution as $p(K) = \frac{\lambda_T^K e^{-\lambda_T}}{K!}$ where λ_T is the expected number of parse graphs in $I_\wedge$. The probability for PG is of the following form

$$p(PG \mid I_\wedge) = \frac{1}{Z} \exp\{-E(PG \mid I_\wedge)\} \tag{11}$$

where Z is the normalization factor and is summed over all PG as $Z = \sum_{PG} \exp\{-E(PG \mid I_\wedge)\}$. The most likely interpretation of $I_\wedge$ can be found by maximizing the following posterior probability

$$PG^* = \arg\max_{PG} p(PG \mid I_\wedge) \tag{12}$$

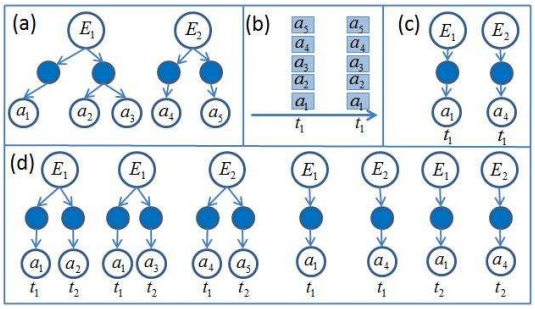

Figure 12: (a) The two T-AOGs of single event E_1 and E_2. (b) The input in the worst case. (c) The parse graphs at time t_1. (d) The parse graphs at time t_2

When the most possible interpretation is obtained, the goal at frame I_T can be inferred as the single event whose parse graph pg_i explains I_T, and the intent can be predicted by the parse graph pg_i.

4.2. Generating parse graphs of single events

We implemented an online parsing algorithm for T-AOG based on Earley's [6] parser to generate parse graphs based on the input data. Earley's algorithm reads terminal symbols sequentially, creating a set of all pending derivations (states) that is consistent with the input up to the current input terminal symbol. Given the next input symbol, the parsing algorithm iteratively performs one of three basic operations (prediction, scanning and completion) for each state in the current state set.

For clarity, we use two simple T-AOGs of E_1 and E_2 without set nodes as shown in Figure12(a) to show the parsing process. Here we consider the worst case, that is, at each time, the input will contain all the atomic actions in E_1 and E_2 as shown in Figure12(b). At time t_0, in the prediction step, E_1' s first atomic action a_1 and E_2' s first atomic action a_4 are put in the open list. At time t_1, in the scanning step, since a_1 and a_4 are in the input, they are scanned in and there are two partial parse graphs at t_1 as shown in Figure 12(c). Notice that we do not remove a_1 and a_4 from the open list. This is because the input is ambiguous, if the input at t_1 is really a_1, then it cannot be a_4 and should not be scanned in and should stay in the open list waiting for the next input. It is the same that if the input at t_1 is really a_4. Then based on the parse graphs, a_2, a_3 and a_5 are predicted and put in the open list. Then at time t_1, we have a_1, a_2, a_3, a_4, a_5 in the open list. At time t_2, all of the five nodes in the open list are scanned in and we will have 7 parse graphs (five new parse graphs plus the two parse graphs at t_1) as shown in Figure 12(d). The two parse graphs at t_1 are kept unchanged at t_2 to preserve the ambiguities in the input. This process will continue iteratively and all the possible parse graphs of E_1 and E_2 will be generated.

4.3. Run-time incremental event parsing

As time passes, the number of parse graphs will increase rapidly and the number of the possible interpretations of the input will become huge, as Figure 13(a) shows. However, the number of acceptable interpretations (PG with probability higher than a given threshold) does not keep increasing, it will fluctuate and drop sharply at certain time, as shown in Figure 13(b). We call these time points the "decision moments". This resembles human cognition. When people watch others taking some actions, the number of possible events could be huge, but at certain times, when some critical actions occurred, most of the alternative interpretations can be ruled out.

Our parsing algorithm behaves in a similar way. At each frame, we compute the probabilities of all the possible interpretations and only the acceptable interpretations are kept. The parse graphs which are not contained in any of these acceptable interpretations are pruned. This will reduce the complexity of the proposed algorithm greatly.

4.4. Multi-agent Event parsing

When there are multiple agents in the scene, we can do event parsing for each agent separately. That is, for each agent in the scene, the atomic actions are detected (all other agents are regarded as objects in the scene) and parsed as mentioned above, then the interpretations of all the agents in the scene are obtained.

5. Experiments

5.1. Data set

For evaluation, we collect videos in 5 indoor and outdoor scenes, including office, lab, hallway, corridor and near vending machines. Figure 14 shows some screen-shots of the videos. The training video total lasts for 60 minutes, and contains 34 types of atomic actions (26 of the 34 types of atomic actions are listed in Table 2 for the office scene) and

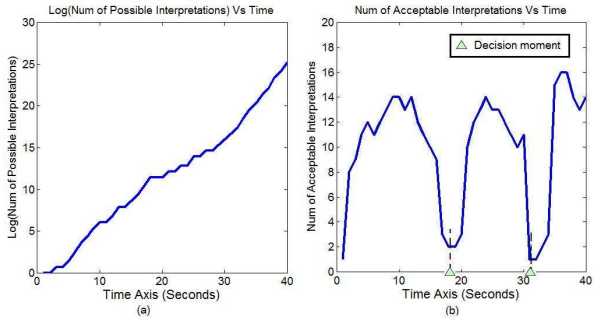

Figure 13: (a) The number of possible interpretations (in logarithm) vs time (in seconds). (b) The number of acceptable interpretations vs time. The decision moments are the time points on which the critical actions happen and the number of acceptable interpretations drops sharply.

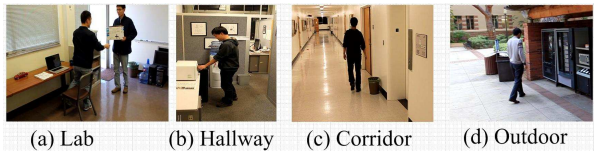

Figure 14: Some screen shots of the data.

12 events categories. Each event happens 3 to 10 times.

The structures of the T-AOG are learned automatically from the training data as described in section 3, the parameters and temporal relations are also learned from the training data. The testing video lasts 50 minutes and contains 12 event categories, including single-agent events like getting water and using a microwave, and multi-agent events like discussing at the white board and exchanging objects. The testing video also includes event insertion such as making a call while getting water.

5.2. Event recognition

The performance of event recognition is shown in Table 4. Figure 15 shows the recognition results of events which may involve multiple agents and happen concurrently.

Using the learned T-AOG, we parse the sequence of atomic actions extracted from a long video in Figure16. The sequence is already compressed so that repeating subsequences are suppressed into single symbols. In the zoomed-out parts of the parse graph in Figure16, we also show the detected bounding boxes of the agent. The semantic description for different non-terminal nodes is also illustrated.

5.3. Goal inference and intent prediction

Besides the classification rate, we also evaluate the precision of the goal inference and intent prediction online. We compare the result of the proposed algorithm with 5 human subjects as was done in the cognitive study with toy examples in a maze world in [2]. The participants viewed the videos with several judgement points, at each judgement point, the participants were asked to infer the goal of the agent and predict his next action with probability.

Table 4: Recognition accuracy of our algorithm.

Scene	Number of event instances	Correct	Accuracy
Office	32	29	0.906
Lab	12	12	1.000
Hallway	23	23	1.000
Corridor	9	8	0.888
Outdoor	11	11	1.000

Figure 17 (a) shows five judgement points of an event insertion (making a call in the process of getting water). Figure 17 (b) shows the experimental results of event segmentation and insertion. Figure 17 (c) shows the goal inference result obtained by participants and our algorithm respectively, and Figure 17 (d) shows the intent prediction results. Our algorithm can predict one or multiple steps according to the parse graph. Here we only show the result of predicting one step. Although the probabilities of the goal inference and intent prediction results are not the same as the average of the participants, the final classifications are the same. In the testing video, we set 30 judgement points in the middle of events. The accuracy of goal inference is 90% and the accuracy of intent prediction is 87%.

5.4. Atomic action recognition with event context

Due to the ambiguity of bottom up detection, the sequence of detected atomic actions is noisy and prone to error. We propose to use the learned T-AOG to "de-noise" the atomic actions sequence. With the learned spatial and temporal grammars as the prior, the detection of atomic actions follows

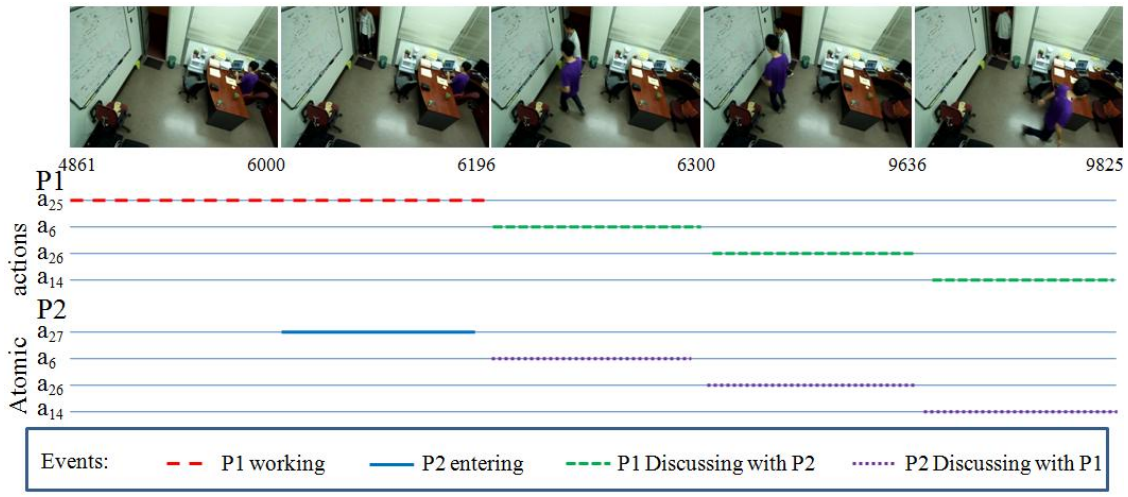

Figure 15: Experiment results of event parsing for multiple agents. Agent P1 works during frames 4861 to 6196, agent P2 enters the room from frames 6000 to 6196, then they go to the white board, have a discussion and leave the board. The semantic meaning of the atomic actions are shown in Table 2.

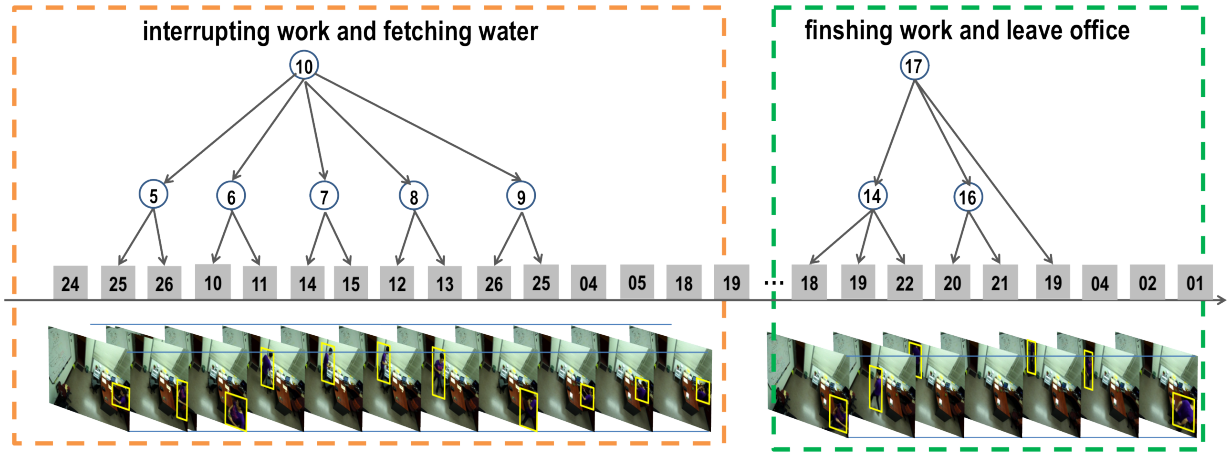

Figure 16: Video parsing result.

a Bayesian maximum-a-posteriori:

$$\mathbf{a}^* = \arg\max_{\mathbf{a}} p(\mathbf{r}|\mathbf{a};\Theta)p(\mathbf{a};\mathcal{G})$$

where $\mathbf{r}$ is the sequence of grounded relations in the video. It is more robust than merely using bottom up proposals:

$$\mathbf{a}_{\text{bottom up}} = \arg\max_{\mathbf{a}} p(\mathbf{r}|\mathbf{a};\Theta)$$

where $\mathcal{G}$ is the learned grammar, and Θ are parameters of the bottom up detectors of atomic actions. We perform an experiment on a collection of 12061 frames.

Figure 18 shows the ROC curve of the recognition results of all the atomic actions in the testing data. The ROC is computed by changing the threshold used in the detection of atomic actions. From the ROC curve we can see that with event context, the recognition rate of atomic actions is improved greatly.

5.5. Scene labeling using events

In the previous sections, the learning and parsing of T-AOG relies on the detection or manual labeling of objects in the scene. Now we try to release this requirement of manual labeling, and use the T-AOG to infer scene semantics automatically, thus closing the loop of unsupervised learning.

Our objective is to label the scene image, especially objects involved in the event parsing for a

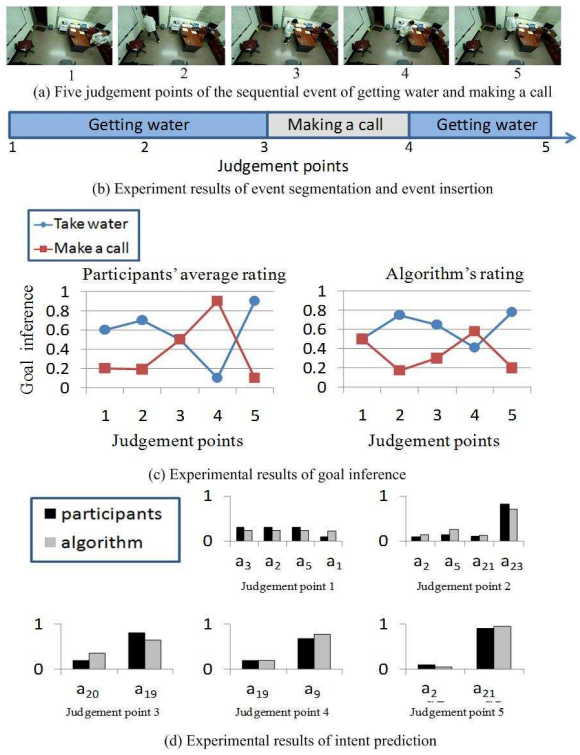

Figure 17: Experiment results of event segmentation, insertion, goal inference and intent prediction. The semantic meanings of the atomic actions in (d) are shown in Table 2.

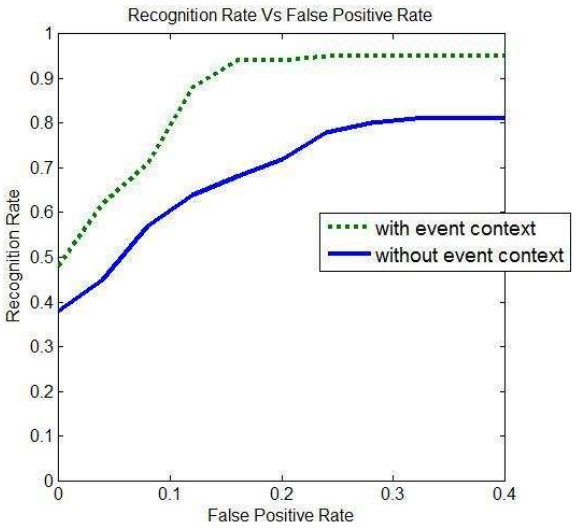

Figure 18: The ROC curve of recognition results of atomic actions.

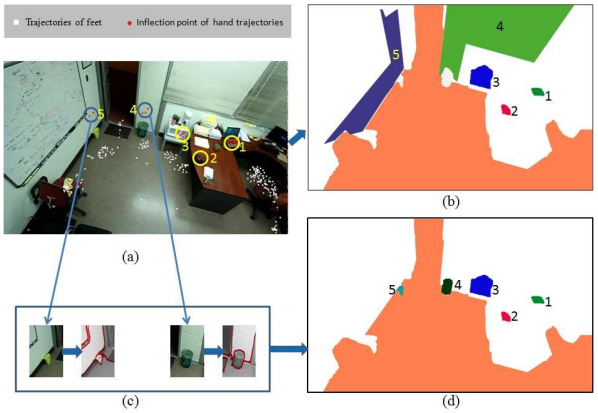

Figure 19: Scene labeling by parsed trajectories. (a) The trajectories of the agent's hands and feet. (b) The segmentation of objects by the trajectory "scribbles". (c) The segmentation of adjacent areas of 4 and 5. (d) The final labeling result for interesting objects.

video $\mathbf{I}_\Lambda$ in a time interval $\Lambda = [0, T]$. For example, a drinking action indicates the location of a cup, while a sitting function indicates a chair.

Suppose pg is an event parse graph from $\mathbf{I}_\Lambda$, and

$$R(\text{pg}) = \{r_1, ..., r_N\}$$

is the set of relations in pg for contextual actions involving the interactions between agent at observed position x_i^{agt}, and object at unknown position x_i^l where l is the object label: $l \in \Omega_L = \{'\text{desk}', '\text{chair}', '\text{cup}', ...\}$. Thus $r_i = r_i(x_i^{\text{agt}}, x_i^l)$. This can be easily extended to multi-way relations. We denote by $L = \{L(x), L(x) \in \Omega_L, \forall x \in rmimagelattice\}$ the scene label.

The scene labeling problem is then formulated as a joint inference,

$$(L^*, \text{pg}^*) = \arg\max p(L, \text{pg}^* | \mathbf{I}_\Lambda)$$

where we denote pg $= (R, \text{pg\_}$, R is the alphabet of the T-AOG and pg_ is the hierarchical parse graph. Or,

$$\begin{aligned}(L^*, R^*, \text{pg\_}) &= \arg\max p(L, R, \text{pg\_} | \mathbf{I}_\Lambda) \\ &= \arg\min E(L, R) + E(R, \text{pg\_}) \\ &\quad + E(L) + E(\text{pg\_})\end{aligned}$$

where $E(L)$ is a smoothness prior on L. $E(L, R)$ is the energy terms involving the set of relations R and the object label L,

$$E(L, R) = \sum_{i=1}^{N} K(x_i^l - x_i^{\text{agt}}),$$

with $K()$ being a distance function, and x_i^l being the point that satisfies two conditions:

1. $x_i^l \in \Omega^l = \{x : L(x) = l\}$;
2. It is close to x_i^{agt}.

So

$$x_i^l = \arg\min_{x_i \in \Omega_l} K(x_i - x_i^{\text{agt}}).$$

Thus the labeling component given pg or $R(\text{pg})$ is

$$L^*|_{\text{pg}} = \arg\min E(L, R) + E(L).$$

The first time is similar to the "user scribbles" in interactive segmentation and labeling [33]. Each label $l \in \Omega_L$ has a set of scribble points $\{x_j^l : j = 1, ..., n_l\}$, where $\sum_{l \in \Omega_L} n_l = N$. The second term utilizes the "scribble" and label the whole scene based on image properties and smoothness assumptions. In Figure 19 we show an example of applying the above scene labeling inference procedure.

Figure 19 (a) shows the trajectories of the agent's hands and feet. Figure 19 (b) shows the segmentation result by the trajectories. The ground is successfully segmented by the trajectories of the feet. The keyboard, phone, microwave are segmented by concentrated trajectories of hands.

The segments 4 and 5 in Figure 19 (b) are too large to be interest objects, so we prune them.

Figure 19 (d) shows the final segmentation result of interesting objects in the scene.

6. Discussion and Conclusion

In summary, we propose a prototype system for event learning, which explores all activities that happen in a certain environment, and organizes them in a meaningful way by a hierarchical event dictionary and a stochastic T-AOG. The learned T-AOG can be used to parse newly observed videos to recognize events. We also show a promising application where it is used to discover scene semantics without manual labeling of the scene. We are working towards applying to more diverse data sets and obtaining richer T-AOG.

We present an algorithm for parsing video events with goal inference and intent prediction. Our experiments results show that events, including those involving multi-agents and those happening concurrently can be recognized accurately, and the ambiguity in the recognition of atomic actions can be reduced largely using hierarchical event contexts.

Future work. The objects of interest in the scene are detected semi-automatically at present. The event context provides a lot of information of the objects involved in the event, and can be utilized to detect and recognize objects. We are actively pursuing further progress in the following aspects:

- Using kinect data to better define agent poses in the 3D setting.
- Clustering more specific actions. An action can be defined as a set of typical configurations, each specified by a number of spatial interactions between agent and environment. E.g. a sitting pose can be specified by interactions between person's body and chair, hand and keyboard, body and desk etc.
- Using n-nary relations to handle group activities.

Acknowledgement

This work is done when Dr. Mingtao Pei is a research fellow at UCLA. We thank the support of NSF grant IIS-1018751, ONR MURI grant N00014-10-1-0933 and DARPA MSEE project FA 8650-11-1-7149 at UCLA. The first author author thanks the support of NSF of China grant 90920009.

References

[1] G. Csibra, G. Gergely, Obsessed with goals: Functions and mechanisms of teleological interpretation of actions in humans, Acta Psychologica (2007) 60–78.

[2] C. L.Baker, R. Saxe, J. B.Tenenbaum, Action understanding as inverse planning, Cognitions (2009) 329–349.

[3] H. Chen, Z. J. Xu, Z. Q. Liu, S.-C. Zhu, Composite templates for cloth modeling and sketching, in: CVPR, 2006.

[4] S.-C. Zhu, D. Mumford, A stochastic grammar of images, Foundat. Trends Comput graphics Vision 2 (2007) 259–362.

[5] A. Gupta, P. Srinivasan, J. Shi, L. S. Davis, Learning a visually grounded storyline model from annotated videos, in: CVPR, 2009.

[6] J. C. Earley, An efficient context-free parsing algorithm, Ph.D. thesis, Carnegie Mellon Univ. (1968).

[7] F. Han, S.-C. Zhu, Bottom-up/top-down image parsing with attribute grammar, PAMI 31 (1) (2009) 59–73.

[8] M. Brand, N. Oliver, A. Pentland, Coupled hidden markov models for complex action recognition, in: CVPR, 1997.

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

[10] K. Otsuka, J. Yamato, Y. Takemae, H. Murase, Conversation scene analysis with dynamic bayesian network based on visual head tracking, in: ICME, 2006.
[11] M. Al-Hames, G. Rigoll, A multi-modal mixed-state dynamic bayesian network for robust meeting event recognition from disturbed data, in: ICME, 2005.
[12] M. S. Ryoo, J. K. Aggarwal, Recognition of composite human activities through context-free grammar based representation, in: CVPR, 2006.
[13] Y. A. Ivanov, A. F. Bobick, Recognition of visual activities and interactions by stochastic parsing, PAMI 8 (2000) 852 – 872.
[14] S. W. Joo, R. Chellappa, Recognition of multi-object events using attribute grammars, Processing of international conference of image process (2006) 2897 – 2900.
[15] Z. Zhang, T. Tan, K. Huang, An extended grammar system for learning and recognizing complex visual events, PAMI (2011) 240–255.
[16] A. Hakeem, Y. Sheikh, M. Shah, Casee: A hierarchical event representation for the analysis of videos, in: The Nineteenth National Conference on Artificial Intelligence, 2004.
[17] B. Georis, M. Maziere, F. Bremond, M. Thonnat, A video interpretation platform applied to bank agency monitoring, in: Proc. 2nd Workshop of Intelligent Distributed Surveillance System, 2004.
[18] M. Albanese, R. Chellappa, V. Moscato, V. Subrahmanian, Pads: A probabilistic activity detection framework for video data, PAMI (2010) 2246 – 2261.
[19] J. Yuan, Y. Wu, M. Yang, From frequent itemsets to semantically meaningful visual patterns, in: SIGKDD, 2007.
[20] F. Nater, H. Grabner, L. V. Gool, Exploiting simple hierarchies for unsupervised human behavior analysis, in: CVPR, 2010.
[21] D. Moore, I. Essa, Recognizing multitasked activities using stochastic context-free grammar, in: AAAI, 2001.
[22] M. Pei, Y. Jia, S.-C. Zhu, Parsing video events with goal inference and intent prediction, in: ICCV, 2011.
[23] Z. Si, M. Pei, B. Yao, S.-C. Zhu, Unsupervised learning of event and-or grammar and semantics from video, in: ICCV, 2011.
[24] A. Torralba, K. P. Murphy, W. T. Freeman, Sharing visual features for multiclass and multiview object detection, PAMI (2007) 854 – 869.
[25] Y. Zhao, S.-C. Zhu, Image parsing with stochastic scene grammar, in: NIPS, 2011.
[26] M. Rodriguez, S. Ali, T. Kanade, Tracking in unstructured crowded scenes, in: ICCV, 2009.
[27] B. Yao, S.-C. Zhu, Learning deformable action templates from cluttered videos, in: ICCV, 2009.
[28] J. F. Allen, G. Ferguson, Actions and events in interval temporal logic, Journal of Logic Computation 4.
[29] Y. N. Wu, Z. Si, H. Gong, S.-C. Zhu, Learning active basis model for object detection and recognition, IJCV 90 (2) (2010) 198–230.
[30] S. D. Pietra, V. D. Pietra, J. Lafferty, Inducing features of random fields, PAMI 19 (4) (1997) 380–393.
[31] S.-C. Zhu, Y. N. Wu, D. B. Mumford, Minimax entropy principle and its applications to texture modeling, Neural Computation 9 (8) (1997) 1627–1660.
[32] K. Tu, V. Honavar, Unsupervised learning of probabilistic context-free grammar using iterative biclustering, in: Proceedings of 9th International Colloquium on Grammatical Inference (ICGI 2008), 2008.
[33] Y. Zhao, S.-C. Zhu, S. Luo, Co3 for ultra-fast and accurate interactive segmentation, in: ACM Multimedia, 2010.