Parsing Video Events with Goal inference and Intent Prediction

Mingtao Pei[a,b], Yunde Jia[a], Song-Chun Zhu[b]
[a] Lab of Intelligent Info. Technology, Beijing Institute of Technology
[b] Department of Statistics, University of California, Los Angeles
{peimt,jiayunde}@bit.edu.cn, sczhu@stat.ucla.edu

Abstract

In this paper, we present an event parsing algorithm based on Stochastic Context Sensitive Grammar (SCSG) for understanding events, inferring the goal of agents, and predicting their plausible intended actions. The SCSG represents the hierarchical compositions of events and the temporal relations between the sub-events. The alphabets of the SCSG are atomic actions which are defined by the poses of agents and their interactions with objects in the scene. The temporal relations are used to distinguish events with similar structures, interpolate missing portions of events, and are learned from the training data. In comparison with existing methods, our paper makes the following contributions. i) We define atomic actions by a set of relations based on the fluents of agents and their interactions with objects in the scene. ii) Our algorithm 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; iii) The algorithm infers the goal of the agents and predicts their intents by a top-down process; iv) The algorithm improves the detection of atomic actions by event contexts. We show satisfactory results of event recognition and atomic action detection on the data set we captured which contains 12 event categories in both indoor and outdoor videos.

1. Introduction

Cognitive studies[8] 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[7] whose actions are often planned to achieve certain goals. In this way, we can interpret observed events in terms of inferring the underlying goals and predicting the next actions.

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

To achieve the above event understanding capability, we need to solve several problems: i) Representing events in hierarchical structure with temporal relations. ii) Dealing with event insertions, interruptions, multi-agent events and agent-object interactions. iii) Preserving the ambiguities both in the lower level atomic action detection and higher level event recognition to achieve globally optimized solution.

Existing methods for event representation and recognition can be divided into two categories. 1) HMMs and DBN based methods. Brand et al.[5] modeled human actions by coupled HMMs. Natarajan[14] described an approach based on Coupled Hidden Semi Markov Models for recognizing human activities. Kazuhiro et al [15] built a conversation model based on dynamic Bayesian network. Al-Hames and Rigoll [2] 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 [13] used the context free grammar (CFG) to model and recognize composite human activities. Ivanov and Bobick [10] proposed a hierarchical approach using a stochastic context free grammar (SCFG). Joo and Chellappa [17] used probabilistic attribute grammars to recognize multi-agent activities in surveillance settings. Zhang et al [19] applied an extended grammar approach to modeling and recognizing complex visual 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

2011 IEEE International Conference on Computer Vision
978-1-4577-1102-2/11/$26.00 ©2011 IEEE

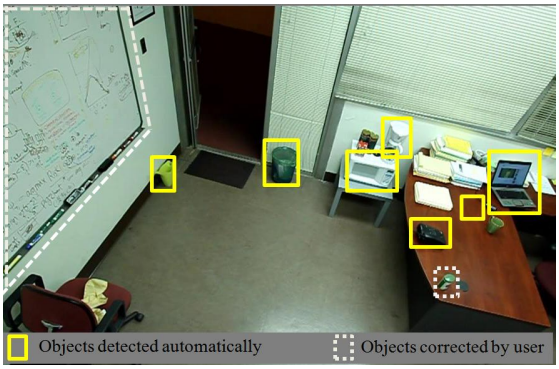


Figure 1. The Detection result of objects in an office scene, the objects of interest include cup, phone, laptop, trash-can, bucket, tea box, microwave, dispenser and white board. The tea box and the white board can not be detected automatically and are corrected by user.

VERL[1, 4], and PADS[3].

In this paper, we represent events by Stochastic Context Sensitive Grammar (SCSG). The SCSG is embodied in an And-Or Graph (AoG) which was first introduced to compute vision in [6] and [20], and has been used in [9] to analyze sports videos. The AoG represents the hierarchical decompositions from events to sub-events to atomic actions. Leaf nodes for atomic actions are defined by a set of fluents (time varying properties) of the agents and their interactions with objects in the scene. And-nodes represent the composition and routine (chronological order) of their sub-nodes. Or-nodes represent the alternative ways to realize a certain event, where each alternative has an associated probability to account for its branching frequency.

In contrast to HMMs and DBN, in the AoG, the And-nodes express long time sequence and enforce the higher order constraints than HMMs, the Or-nodes enable the reconfiguration of the structures, and thus are more expressive that the fix-structured DBN. The AoG also represents the temporal relations between multiple sub-events by the horizontal links between the nodes, by which, the temporal relations are better utilized than they are in CFG and SCFG.

The contributions of our paper are: 1) We represent the atomic actions by a set of relations. 2) Our 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 posterior probability. 3) The agent's goal and intent at each time point is inferred by bottom-up and top-down process based on the parse graphs contained in the most probable interpretation. 4) We show that the detection result of atomic actions can be improved by event contexts.

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 result of the algorithm are evaluated by human subjects and our experiments show satisfactory results.

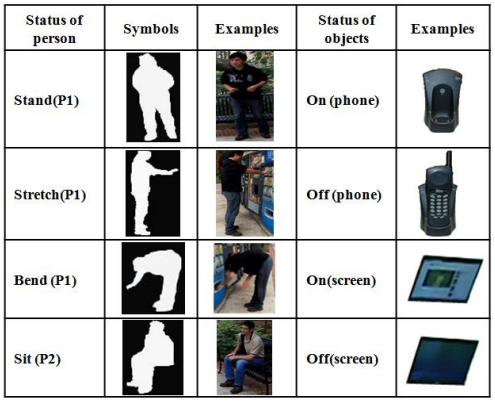


Figure 2. The unary relations or fluents. The left are four fluents of agent: 'Stand', 'Stretch', 'Bend' and 'Sit'. The right shows two fluents ('On' and 'Off') of a phone and a laptop screen.

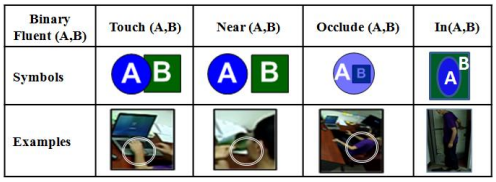


Figure 3. The binary spatial relations between agents (or parts) and background objects.

2. Event representation by AoG

The AoG includes And-nodes, Or-nodes, leaf nodes and higher order temporal relations. The structures of the AoG are learned automatically from the training data as in our companion paper [16], the parameters and temporal relations are also learned from the training data.

2.1. Atomic actions — the leaf nodes

We define an atomic action as a set of relations $a = \{r_1, ...r_J\}$ including both unary and binary relations.

- An unary relation $r(A)$ is also called the fluent, i.e. a time varying property of the agent or object A in the scene. As Figure 2 shows, it could be the pose (e.g. stand and bend) and object states (e.g. on and off).

- A binary relation $r(A, B)$ is the spatial relation (e.g. touch and near) between A, B which could be agents, body parts (hands, feet), and objects. Figure 3 illustrates some typical relations.

There are 13 classes of interest objects including cup, laptop, water dispenser in our training and test data. These

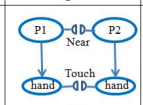
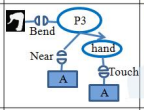
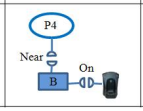
Atomic Actions	Fluents	Symbols		Examples
		Foreground	Background	
Shake Hands(P1,P2)	Near(P1,P2) And Touch (P1.hand, P2.hand)			
Use Dispenser(P3)	Bend(P3) and Near(P3, A) And Touch(P3.hand, A)			
Pick up Phone(P4)	Touch(P4, B) And On(B)			

Figure 4. Examples of atomic actions, each relation is shown by 2 half-circles that are bonded together. For the action of 'shaking hands', A, B are both agents.

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 [18], and the detection result is interactively edited. 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.

We use a background subtraction algorithm to detect the agents and fluent changes of objects, and we use a commercial surveillance system to track the detected agents, both with errors and ambiguities which are to be corrected by hierarchical event contexts. The unary relations of agents and objects are detected by computing the properties of the foreground area such as the aspect ratio and intensity histogram of the bounding box, and a probability is obtained. The binary relations are defined on the positions of A and B. A and B could be agents (parts) and objects. The head and hands of agent are detected by skin color detection in the foreground area. We use the normal distribution as the detection rule of these relations. The parameters of the distribution can be learned from the training data.

When a relation involves an object, the object is tracked until the relation finishes and the new position of the object will be updated.

Figure 4 shows three examples of the atomic actions. Table 1 lists the 28 atomic actions used in the office scene. There are totally $n = 34$ atomic actions in our experiments. 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 relations. An atomic action $a = \{r_1, ..., r_J\}$, has the following probability given frame I_t,

$$p(a \mid I_t) = \frac{1}{Z} \prod_{j=1}^{J} p(r_j) \propto \exp\{-E(a)\} \qquad (1)$$

where $E(a) = -\sum_{j=1}^{J} \log p(r_j)$ is the energy of a and $Z = \sum_{i=1}^{n=34} (p(a) \mid I_t)$ is the normalization factor, over all possible atomic actions.

Given the input video $I_\wedge$ in a time interval $\wedge = [0, T]$, multiple atomic actions are detected with probability at each frame to account for the ambiguities in the relations contained in the atomic actions, for example, the relation 'touch' cannot be clearly differentiated from relation 'near' unless kinect is used. The other reason is the inaccuracy of foreground detection. Fortunately, most of the ambiguities can be removed by the event context, we will show this in the experiment section.

Node Name	Semantic Name	Node Name	Semantic Name
a_1	arrive at phone	a_9	leave phone
a_2	arrive at trash-can	a_{10}	leave trash-can
a_3	arrive at basin	a_{11}	leave basin
a_4	arrive at dispenser	a_{12}	leave dispenser
a_5	arrive at tea box	a_{13}	leave tea box
a_6	arrive at board	a_{14}	leave board
a_7	arrive at laptop	a_{15}	leave laptop
a_8	arrive at microwave	a_{16}	leave microwave
a_{17}	use laptop	a_{18}	read paper
a_{19}	use tea box	a_{20}	use phone
a_{21}	use dispenser	a_{22}	use microwave
a_{23}	bend down	a_{24}	null
a_{25}	work	a_{26}	discuss
a_{27}	enter	a_{28}	exit

Table 1. The atomic action in the office scene which are the terminal nodes in AoG representation.

2.2. Event composition by And, Or & Set nodes

An event category is represented by a 6-tuple $AoG =< S, V_N, V_T, R, \Sigma, P >$. It embodies a stochastic context sensitive grammar (SCSG). 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 its children nodes. These children nodes must occur in certain temporal order. An Or-node points to 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. The Set-node is a special Or-node which can repeat m times, and is associated with a probability $p(m)$ that 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

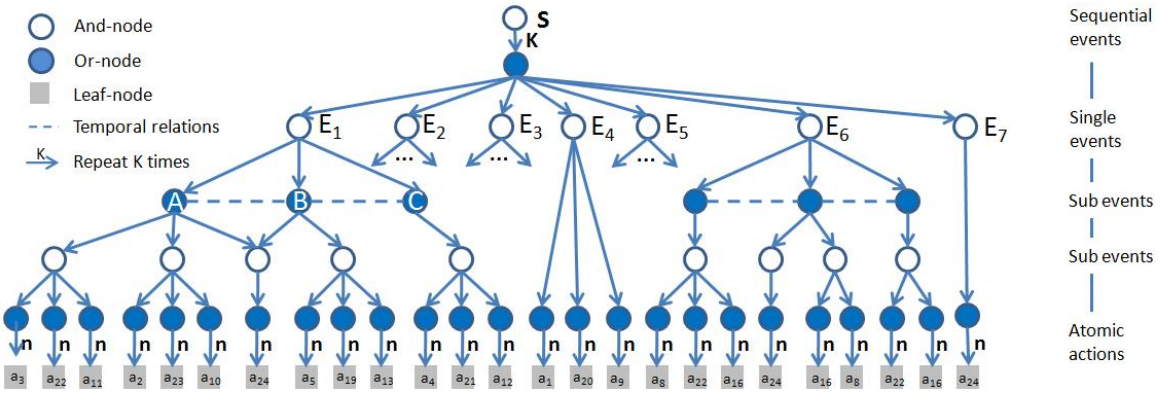


Figure 5. AoG of events in the office scene. S is the root node which represents the sequential events happened in the office. It is a Set-node and could be any combinations of K single events. For example, S could be $E_1|E_2|E_1E_2|E_3E_2E_3|....$ $E_1, ..., E_7$ are And-nodes representing single events. 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.

the events) derivable from the AoG, i.e. its language, and P is the probability model defined on the graph. The AoG of events in the office scene is shown in Figure 5.

2.3. Non-parametric temporal relations

The And-nodes have already defined the chronological order of its sub-nodes, and the Set-nodes representing atomic actions have modeled the lasting time of the atomic action by the frequency of its production rules. In addition, we augment the AoG by adding temporal relations to regulate the durations of nodes in events.

Unlike [10] and [19] which use Allen's 7 binary temporal relations [12], we use non-parametric filters to constrain the durations between multiple nodes. We use the AoG of E_1 in Figure 5 to illustrate the temporal relations. E_1 is an And-node and has three child nodes A, B and C whose durations are τ_A, τ_B and τ_C respectively. For example, an agent bends down in A, does something in B, and stands up in C. The sub-event B could be 'pick up an object' or 'tie shoe laces'. The two cases can only be distinguished from the relative duration τ_B with respect to τ_A, τ_C. Also, when an agent performs event E_1 in a hurry, the durations 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. We denote a temporal filter as $F = (F_1, F_2, F_3)$, and we measure how well the durations $\tau_{E_1} = (\tau_A, \tau_B, \tau_C)$ fits to this filter by their inner product $Tr =< \tau_{E_1}, F >$ in the same way as image filters. The response Tr follows a continuous distribution,

$$p(Tr) \sim h(< \tau, F^* >) \tag{2}$$

Where h is the histogram calculated for Tr from the training data. One may use multiple F to model the relations if needed. The selection of these filters follows the minimum entropy principle [20] that chooses filters telling the most difference between the observed histogram and the synthesis histogram according to the current model.

2.4. Parse graph

A parse graph is an instance of the 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 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.
- $R(pg) = \{Tr_1, ..., Tr_{n(R)}\}$ is the set of temporal relations between the nodes in pg.

The energy of pg is defined as in[20]

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

The first term is the data term, it expresses the energy of the detected leaf nodes (atomic actions) which is computed by eqn 1. The second term is the frequency term, it accounts for how frequently each Or-node decomposes 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 eqn 2.

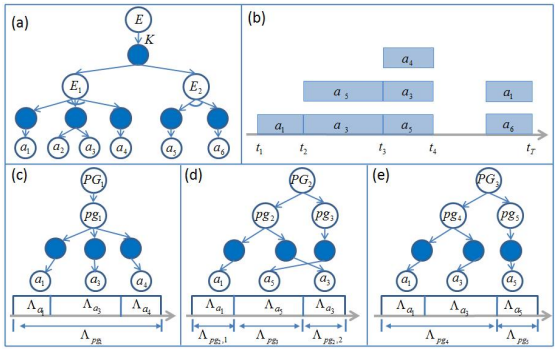


Figure 6. (a) A small AoG. (b) A typical input of the algorithm. (c),(d) and (e) are three possible parse graphs (interpretations) of the video $I_{\wedge[t_1,t_4]}$. Each interpretation segments the video $I_{\wedge[t_1,t_4]}$ into single events at the event level and into atomic actions at the atomic action level.

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,

$$PG = (K, pg_1, ..., pg_K) \tag{4}$$

where K is the number of parse graphs of single event.

3. Event parsing

Firstly we will show the event parsing process by assuming that there is only one agent in the scene in section 3.1 - 3.3. In section 3.4 we will show how to parse events when there are multiple agents in the scene.

3.1. 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 interval of parse graph pg_i. $PG = (K, pg_1, ..., pg_K)$ is regarded as an interpretation of $I_\wedge$ when

$$\begin{cases} \cup_{i=1}^{K} \wedge_{pg_i} = \wedge \\ \wedge_{pg_i} \cap \wedge_{pg_j} = \emptyset \quad \forall ij \quad i \neq j \end{cases} \tag{5}$$

We use a small AoG in Figure 6(a) to illustrate the algorithm. Figure 6(b) shows a sample input of atomic actions. Note that there are multiple atomic actions at each time point. Figure 6(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 6(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 $I_{[t_1,t_4]}$ 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 6(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 6(e) is another parse graph and segments $I_{[t_1,t_4]}$ into two single events E_1 and E_2.

The segmentation of events is automatically integrated in the parsing 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

$$E(PG \mid I_\wedge) = p(K) \sum_{k=1}^{K} (\varepsilon(pg_k \mid I_{\wedge_{pg_k}}) - \log p(k)) \tag{6}$$

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

$$p(PG \mid I_\wedge) = \frac{1}{Z} \exp\{-E(PG \mid I_\wedge)\} \tag{7}$$

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

$$PG^* = \arg\max_{PG} p(PG \mid I_\wedge) \tag{8}$$

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.

3.2. Generating parse graphs of single events

We implemented an online parsing algorithm for AoG based on Earley's [11] 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 AoGs of E_1 and E_2 without set nodes as shown in Figure7(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 Figure7(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 7(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

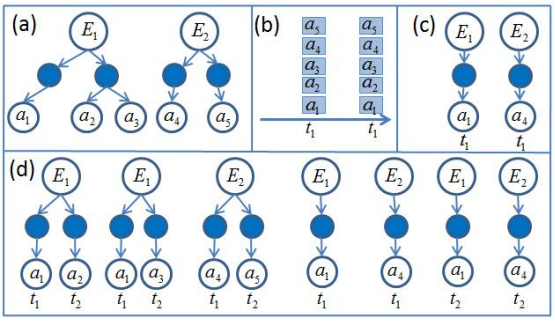


Figure 7. (a) The two AoGs of single event E_1 and E_2. (b) The input in the worst case. (c) The parse graphs at time t_1. (d) The parse graphs at time t_2

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 7(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.

3.3. 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 8(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 8(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.

3.4. 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.

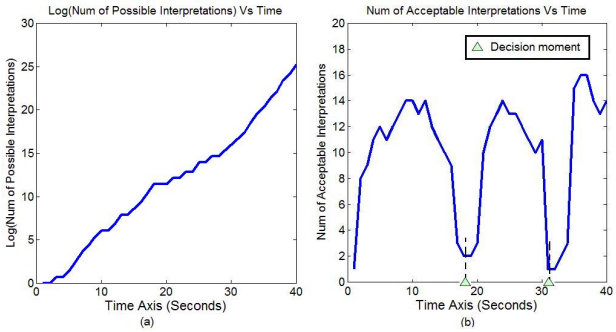


Figure 8. (a) The number of possible interpretations (in logarithm) vs time (in seconds). (b) The number of acceptable interpretations vs time. The decision moments are the time points on which the critical actions happen and the number of acceptable interpretations drops sharply.

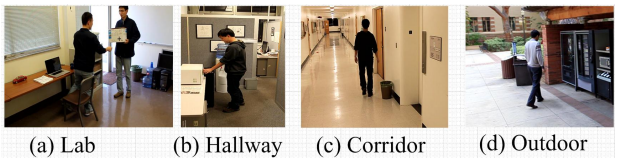


Figure 9. Sample image shots from the collected videos.

4. Experiments

4.1. Data collection

For evaluation, we collect videos in 5 indoor and outdoor scenes, include office, lab, hallway, corridor and around vending machines. Figure 9 shows some screen-shots of the videos. The training video total lasts for 60 minutes, and contains 34 types of atomic actions (28 of the 34 types of atomic actions are listed in Table 1 for the office scene) and 12 events categories. Each event happens 3 to 10 times.

The structures of the AoG are learned automatically from the training data as in our companion paper[16], 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.

4.2. Atomic action recognition with event context

Figure 10 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.

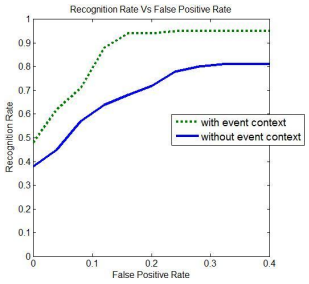


Figure 10. The ROC curve of recognition results of atomic actions.

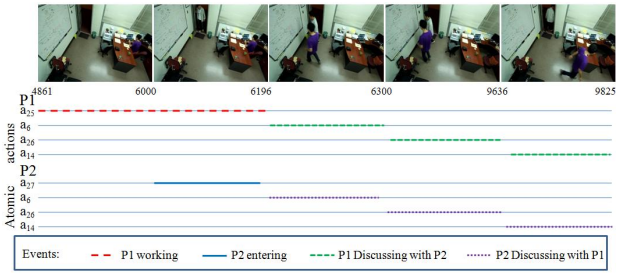


Figure 11. Experiment results of event recognition which involve multiple agents. Agent P1 works during frames 4861 to 6196, agent P2 enters the room from frames 6000 to 6196, then they go to the white board, have a discussion and leave the board. The semantic meaning of the atomic actions could be found in Table 1.

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

Table 2. Recognition accuracy of our algorithm.

4.3. Event Recognition

The performance of event recognition is shown in Table 2. Figure 11 shows the recognition results of events which may involve multiple agents and happen concurrently.

4.4. 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 [7]. 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.

Figure 12 (a) shows five judgement points of an event insertion (making a call in the process of getting water). Figure 12 (b) shows the experimental results of event segmentation and insertion. Figure 12 (c) shows the goal inference result obtained by participants and our algorithm respectively, and Figure 12 (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%.

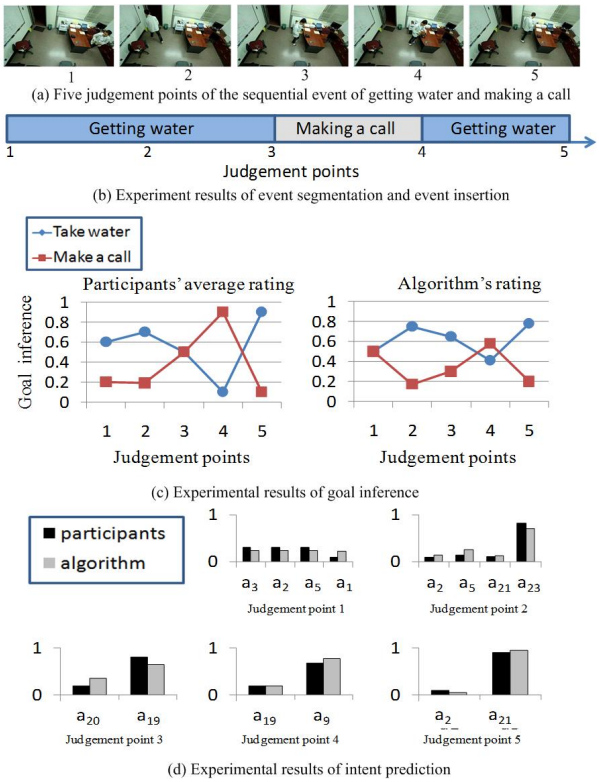
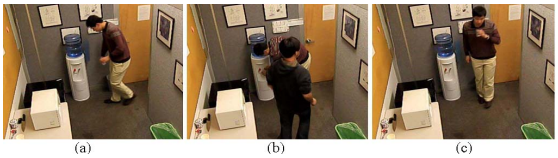


Figure 12. Experiment results of event segmentation, insertion, goal inference and intent prediction. The semantic meaning of the atomic actions in (d) could be found in Table 1.

4.5. Event interpolation

When some atomic actions are not detected because of occlusions or missing detections in the input data, these atomic actions are interpolated as follows: For each predicted atomic action a_i, if it is not detected at the next time point, we will add a_i in the detection results of atomic action with a low probability. After parsing, the missed atomic actions will be interpolated by the event context.

We tested the performance of event interpolation, as shown in Figure 13. During the process of getting water, another agent appeared and occluded the agent, then the atomic action 'use dispenser' can not be detected. By our algorithm, the atomic action 'use dispenser' can be inter-



(a) (b) (c)

Figure 13. (a) (b) and (c) are three representative frame of atomic action 'arrive dispenser', 'use dispenser' and 'leave dispenser' respectively.

Comparison of Algorithms	HMM Based		DBN Based		Grammar Based	
	Brand[6]	Natarajan[14]	Kazuhiro[15]	M.Al-Hames[2]	Ivanov [12]	Ours
Relations over multiple nodes	No	No	No	No	No	Yes
Goal inference and intent prediction	No	No	No	No	No	Yes
Primitive detection with uncertainty	No	No	No	No	Yes	Yes
Primitive correction by context	No	No	No	No	Yes	Yes
Interaction with Objects	No	No	No	No	Yes	Yes
Concurrent events Recognition	No	Yes	Yes	No	No	Yes

Table 3. Qualitative Comparison with Previous Work

polated by event interpolation and the whole event can be recognized as well. In the testing video, there are 9 atomic actions that could not be detected because of viewing angle and occlusion. We also remove 21 detected atomic actions from the detection results to test the performance of event interpolation. By event interpolation, 27 of the 30 missing and occluded atomic actions are "detected" successfully.

5. Conclusion and Future work

We present an AoG representation and an algorithm for parsing video events with goal inference and intent prediction. Our experiments results show that events, including events involve multi-agents and events that happen concurrently can be recognized accurately, and the ambiguity in the recognition of atomic actions can be reduced largely using hierarchical event contexts. In Table 3 we compare the proposed algorithm with previous 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 refer to a companion paper[16] for details of learning the AoG and semantic labeling of small objects in the scenes. As kinect can get 3D information precisely in indoor scene, we plan to use kinect to detect more meticulous atomic actions, and parse more complex events in future work.

Demos and data set are available at http://mcislab.cs.bit.edu.cn/member/∼peimingtao /EventParsing.html.

Acknowledgement This work is done when Pei is a research fellow at UCLA. We thank the support of NSF grant IIS-1018751,ONR MURI grant N00014-10-1-0933 and N00014-11-c-0308 at UCLA. The authors also thank the support of NSF of China grant 90920009.

References

[1] A.Hakeem and M.Shah. Ontology and taxonomy collaborated frame work for meeting classification. *ICPR*, 2004. 2

[2] M. Al-Hames and G. Rigoll. A multi-modal mixed-state dynamic bayesian network for robust meeting event recognition from disturbed data. *IEEE ICME*, 2005. 1

[3] M. Albanese, R. Chellappa, V. Moscato, and V. Subrahmanian. Pads: A probabilistic activity detection framework for video data. *PAMI*, 2010. 2

[4] B.Georis, M.Maziere, F.Bremond, and M.Thonnat. A video interpretation platform applied to bank agency monitoring. *Proc,2nd Workshop of Intelligent. Distributed Surveillance System*, 2004. 2

[5] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. *CVPR*, 1997. 1

[6] H. Chen, Z. J. Xu, Z. Q. Liu, and S. C. Zhu. Composite templates for cloth modeling and sketching. *CVPR*, 2006. 2

[7] R. S. Chris L.Baker and J. B.Tenenbaum. Action understanding as inverse planning. *Cognitions*, 2009. 1, 7

[8] G. Csibra and G. Gergely. Obsessed with goals: Functions and mechanisms of teleological interpretation of actions in humans. *Acta Psychologica*, 2007. 1

[9] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Learning a visually grounded storyline model from annotated videos. *CVPR*, 2009. 2

[10] Y. A. Ivanov and A. F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *PAMI*, 8, 2000. 1, 4

[11] J.C.Earley. *An Efficient Context-Free Parsing Algorithm*. PhD thesis, Carnegie-Mellon Univ, 1968. 5

[12] J.F.Allen and G.Ferguson. Actions and events in interval temporal logic. *Journal of Logic Computation*, 4, 1994. 4

[13] M.S.Ryoo and J.K.Aggarwal. Recognition of composite human activities through context-free grammar based representation. *CVPR*, 2006. 1

[14] P. Natarajan and R. Nevatia. Coupled hidden semi markov models for activity recognition. *IEEE Workshop on Motion and Video Computing*, 2007. 1

[15] K. Otsuka, J. Yamato, Y. Takemae, and H. Murase. Conversation scene analysis with dynamic bayesian network based on visual head tracking. *ICME*, 2006. 1

[16] Z. Z. Si, M. T. Pei, B. Yao, and S. C. Zhu. Unsupervised learning of event and-or grammar and semantics from video. *ICCV*, 2011. 2, 6, 8

[17] S.W.Joo and R.Chellappa. Recognition of multi-object events using attribute grammars. *Processing of international conference of image process*, 2006. 1

[18] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *PAMI*, 2007. 3

[19] Z. Zhang, T. Tan, and K. Huang. An extended grammar system for learning and recognizing complex visual events. *PAMI*, 2011. 1, 4

[20] S.-C. Zhu and D. Mumford. A stochastic grammar of images. *Foundat. Trends Comput graphics Vision*, 2, 2007. 2, 4