Inferring Social Roles in Long Timespan Video Sequence

Jiangen Zhang[1,4], Wenze Hu[2], Benjamin Yao[2], Yongtian Wang[1,3] and Song-Chun Zhu[2,4]
[1]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China 100081
[2]Department of Statistics University of California, Los Angeles, Los Angeles, CA 90095
[3]School of Optics and Electronics, Beijing Institute of Technology, Beijing, China 100081
[4]Lotus Hill Research Institute, Wuhan, China 430060
jiangenzh@bit.edu.cn,{wzhu,zyyao}@stat.ucla.edu,wyt@bit.edu.cn,sczhu@stat.ucla.edu

Abstract

In this paper, we present a method for inferring social roles of agents (persons) from their daily activities in long surveillance video sequences. We define activities as interactions between an agent's position and semantic hotspots within the scene. Given a surveillance video, our method first tracks the locations of agents then automatically discovers semantic hotspots in the scene. By enumerating spatial/temporal locations between an agent's feet and hotspots in a scene, we define a set of atomic actions, which in turn compose sub-events and events. The numbers and types of events performed by an agent are assumed to be driven by his/her social role. With the grammar model induced by composition rules, an adapted Earley parser algorithm is used to parse the trajectories into events, sub-events and atomic actions. With probabilistic output of events, the roles of agents can be predicted under the Bayesian inference framework. Experiments are carried out on a challenging 8.5 hours video from a surveillance camera in the lobby of a research lab. The video contains 7 different social roles including "manager", "researcher", "developer", "engineer", "staff", "visitor" and "mailman". Results show that our proposed method can predict the role of each agent with high precision.

1. introduction

There are roughly two types of visual information that can contribute to recognizing a person's social role: 1) appearance attributes, such as the person's uniform, accessory, etc. 2) behavior pattern that is represented by the person's motions and interactions with other objects (including other human) in the scene. The later case has wide applications in the video surveillance domain where image resolutions are generally low, and/or in environments where casual dressings (instead of uniforms) are common. For example, using a surveillance cameras installed in the lobby of a bank, it would be quite useful to recognize whether a person is a customer or a bank employee by looking at his/her movements pattern. The ability to perform this type of task seems natural to human, as recent studies suggest that infants as young as six months could make complex social attribution inference by observing motion interactions between simple tokens in videos [6]. For this problem, Ullman et al. proposed a hidden Markov model and corresponding algorithms [13], but their experiments are conducted on simple, simulated environments instead of real scenes.

In this paper, we are interested in inferring social roles of agents (persons) from their daily behavior patterns observed in surveillance scenes. To this aim, we collected long timespan video shot in the lobby of a research lab, where each person may appear once or multiple times. Our target is to recognize 7 different social roles presented in the video, including "manager", "researcher", "developer", "engineer", "staff", "visitor" and "mailman". To our best knowledge, this kind of role inference task has yet been well studied by researchers in the computer vision community. Some related efforts have recently been made in discovering social connections between a group of person, for example, Ding et al. [5] proposed a method to construct social networks to identify social groups' and groups leaders of the characters in movie. Yu et al. [14] also use social network models, aiming to discover groups of persons in surveillance videos. Although these methods can be used to cluster agents by their attributes inherent in their activities, they do not model the high-order relations of interactions, and thus cannot give an understanding of the video. Besides, the interaction between people used by their approaches is difficult to extract by current tracking algorithms. We propose to use interactions between agents and the environment, which is more reliable in practice since configurations of the scene are static.

In this paper, we study the following problems: 1) Is it possible to infer social roles of agents by observing their

activities in certain scene? 2) How do multiple activities of an agent help infer his/her social role? 3) What is the scene information that can be extracted to help understand the roles of agents?

To solve these problems, we introduce a representation and a model to infer the roles of agents from their trajectories. From short to long timespan, a trajectory is hierarchically explained by atomic actions, sub-events and events. Atomic actions are the smallest entities which can be observed directly in the video. Together with their temporal dependencies, possible events, sub-events and atomic actions are hierarchically organized by an And-or structure, so that a large number of events can be constructed by reusing sub-events and atomic actions. The role of an agent is then represented by another hidden layer that drives the agent to perform a set of events.

By defining compositional rules, grammar models can be naturally used to decompose the trajectories into semantic meanings of atomic-actions, sub-events and events. Using the roles of agents as hyper-parameters, the total number of events conducted by an agent and the proportion of these events are modelled by Poisson and Multinomial distributions respectively. This defines an event generating process that generates an event set by the role of an agent. This event generating process together with the grammar model form a straightforward multi-layer generative model, which starts from the abstract concept of social role to specific atomic actions that are observed from the agent's trajectories.

In experiments, we show that for the lobby scene of research lab data, our approach can achieve an overall role recognition rate of 87.2%, which suggests the potential of our approach in solving role recognition problems. Numerical experiments show that observing up to five events for an agent dramatically reduces the uncertainly of our algorithm, and with more evidence helps, the ambiguity reduced slightly and slowly.

The contributions of this paper are three-folds: 1) We propose a hierarchical representation of events for the purpose of role inference. Existing event representations can be roughly divided into two categories, HMM based methods [2, 9, 1], and grammar based methods [8, 11, 15]. Our representation belongs to the later category. Our event understanding is based on the Earley [7, 12] parsing algorithm. 2) By the And-or structure, we define grammars that can compose a large set of possible events from a small set of atomic actions. The And-or structure used in this paper is a general knowledge representation framework [10], and different variants can be seen in image understanding [16, 3] and machine learning [4]. The And-or structure used here can be considered as a specialization of this framework on video representation. Compared to the And-or Graph [16], we added temporal relations to represent the time dependency of accomplishing sub-events after finishing other sub-events. We further added repetitive edges to represent multiple children nodes for different time stamps by one parent node. 3) We propose a promising approach to solve the role inference problem, a new task that has not been studied in the vision literature.

2. Representation

In this section, we introduce the definitions and representations for the building blocks of our model, including: 1) *Semantic hotspots*, which are locations of semantic importance in a surveillance scene (e.g. "entrance", "exit", "reception desk", etc.); 2) *Atomic actions*, each of which stands for a spatial-temporal interaction between an agent and a semantic hotspot within a short time span (e.g. entering the door", "approaching the desk", etc.); 3) *Events*, which are temporal combinations of atomic actions and commonly have high level semantic meaning (e.g. "mail delivery").

2.1. Semantic Hotspots

We use trajectories of agents in the video sequence to automatically discover semantic hotspots in the scene. For each frame, we extract the feet location of an agent, which is the bottom point on the central axis of the object bounding box, and treat it as a point on the trajectory. Therefore, a trajectory tr of an agent is defined as the sequence of the feet positions at consecutive frames.

The left panel of Fig.1 illustrates sample trajectories from the research lab scene we study. We can see that agents enter and exit the scene from a limited set of positions, which correspond to the entrances and exits of the scene (denoted as source/sink in the right panel of Fig.1). It is also worth noting that trajectories sometimes turn sharply near certain locations, indicating that agents could be doing something there. If we plot the trajectories in 3D (x-y-t), it is clear that agents usually stay at those turning points for a period of time, because, presumably, they are interacting with the semantic objects in the scene.

There is no question that these semantic objects together with the entrances/exits points bears crucial information to our understanding of the video. Therefore, we denote these position as semantic hotspots. Specifically, starting and ending positions of trajectories are called *source/sink hotspots*, and positions where people stay are called *stay hotspots*. The right panel of Fig.1 illustrates the semantic hotspots of the research lab scene. According to the actual scene, we also list the semantic interpretations of each hotspot in the left most column of the Table 1.

To derive these hotspots from trajectories of agents, we first extract the start, end and stay points on each trajectory. It is straightforward to get start/end points of a given trajectory. To extract stay points, we compute the variance of position in a local temporal window at each frame, and chose those points whose variances are below a threshold ϵ.

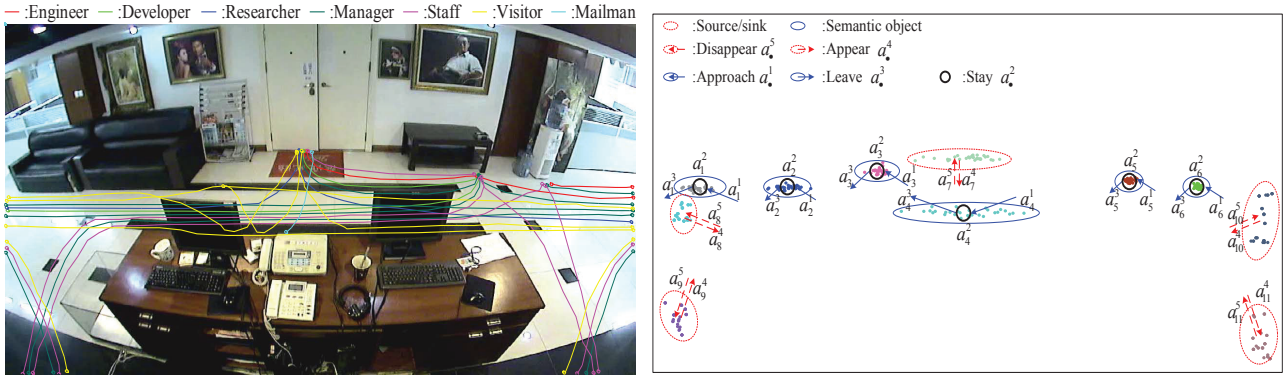

Figure 1. Left: Sample trajectories from all 7 roles in the lobby Scene. Right: start, end and stationary of trajectories are clustered into semantic hotspots using the algorithm specified in Section 2.1. Atomic actions defined in Table 1 (except a^6) are also marked out.

Relations	Approach	Stay	Leave	Appear	Disappear	Null
Sofa 1	a^1_1	a^2_1	a^3_1			
Sofa 2	a^1_2	a^2_2	a^3_2			
Paper stand	a^1_3	a^2_3	a^3_3			
Reception	a^1_4	a^2_4	a^3_4			
Time puncher	a^1_5	a^2_5	a^3_5			
Water dispen.	a^1_6	a^2_6	a^3_6			
Door				a^4_7	a^5_7	
UpLeft exit				a^4_8	a^5_8	
LowLeft exit				a^4_9	a^5_9	
UpRight exit				a^4_{10}	a^5_{10}	
LowRight exit				a^4_{11}	a^5_{11}	
Anywhere else						a^6

Table 1. List of defined atomic actions. For an atomic action a^j_i, the subscript $i \in \{1, 2, ..., 11\}$ is the ID of a hotspot, the superscript $j \in \{1, 2, ..., 5\}$ is the type of relation. Also see Fig.1.

Then we use an EM clustering algorithm to find the cluster centres of these points, which corresponds to the semantic hotspots in the scene. As illustrated in the right panel of Fig.1, each hotspot is the cluster center of a set of points.

2.2. Atomic actions

Using the semantic hotspots derived from the previous section, we define an atomic action by a specific set of spatial and temporal relations between an agent and a semantic hotspots within a shot time span. Each of the relations is associated with a probability model. Then the probability of an atomic action is defined by the product of probabilities from all relations. Our definition of action is similar to [11].

We define four types of relations: *appear*, *disappear*, *distance*, and *duration*. The first three are spatial relations, evaluating the position difference from start/end/during position of a trajectory to a hotspot respectively. The last one is used to evaluate the timespan that an agent is using an object or staying near it. We use a Gaussian distribution as a probability model for these four relations. For example, the probability of distance relation between the feet f of an agent and a hotspot h is defined as:

$$p(\mathbf{x}_f, \mathbf{x}_h) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp(E)$$
$$E = -\frac{1}{2}(\mathbf{x}_f - \mathbf{x}_h - \mu)'\Sigma^{-1}(\mathbf{x}_f - \mathbf{x}_h - \mu) \quad (1)$$

where $\mathbf{x}_f$ and $\mathbf{x}_h$ denote the location of the feet and the hotspot respectively, and μ and Σ are parameters that are fitted from training data by MLE.

With these four relations, we are able to define five types of atomic actions. According to the hotspot's type, the five atomic actions can be classified into two groups: 1) *Enter/exit* a source/sink hotspot, which use the appear and disappear relation between the feet of the agent and the source/sink hotspot. 2) *Approach/use/leave* a stay hotspot. "Use" action is defined by two relations: during relation of agent with a stay hotspot, and distance relations from the hotspot to the feet of a agent. "Approach" and "leave" are defined using distance between feet and the stay hotspot. Although using the same relation, the difference between approach and leave is distinguished by its time dependency on the use action. For these five types, the number of possible atomic actions in a scene is $2n_i + 3n_d$, where n_i is the number of source/sink hotspots and n_d is the number of stay hotspots.

Between transitions of atomic actions or some special trajectories, there could be frames that do not belong to the five categories of atomic actions defined above, in this case, we use an atomic action NULL to explain these frames. This atomic action essentially correspond to a background model in image modelling. Probability of this atomic action is set to be a very low constant (0.1 in our case).

2.3. Event Composition by And-or Structure

The atomic actions define the bottom layer of our And-or structure as shown in Fig.2, by n repetitions, they become atomic action sets that compose sub-events: $s_i = \{a_{it}\}_{t=1}^{\tau}$,

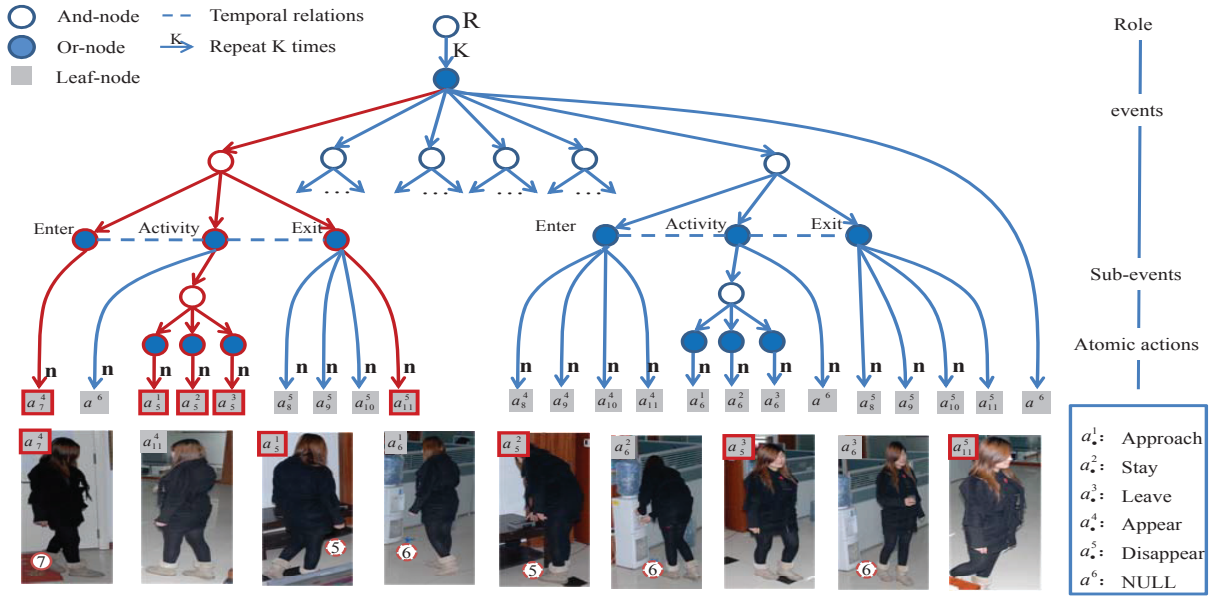

Figure 2. An example And-or structure for the scene shown in Fig.1. All nodes and edges compose an And-or structure, while the node and edges in red are an example parse graph corresponding to an event using time puncher. Some leaf nodes, and sub-event nodes are plotted twice for better illustration. Example leaf nodes which are atomic actions are shown in bottom.

where τ is the timespan of a sub-event, a_i is an atomic action as shown in Table.1 and a_{it} is the atomic action at time t. Sub-events are categorized into three classes: *enter, exit* and *activity*. The enter and exit sub-events are simple and only contain one enter or exit atomic action set. The activity sub-event contains three time dependent atomic action sets or only the NULL atomic action. For example, the activity sub-event of fetching water is composed by atomic action sets of approaching, using and leaving stay hotspot h_6 (corresponds to water dispenser in the scene), which happen one after another.

In the same analogy, an event is defined by the time-dependent composition of sub-events. In our implementation, an event is fixed to have only one enter, one activity and one exit sub-event. While this definition might ignore the cases that an agent performs two or more activity sub-events in one trajectory, we found these cases are very rare in practice, and adding these cases improves the recognition rate very little, but makes the model much more complex. So, with a total of $2n_i + 3n_d + 1$ atomic actions, there are $2n_i + n_d + 1$ possible sub-event categories and $n_i^2 \cdot (n_d + 1)$ possible event categories. Since there are 5 source/sink hotspots and 6 stay hotspots int the scene shown in Fig.1, a total of 175 possible events can be represented by 17 sub events and 29 atomic actions.

3. Role Modeling

Our And-or structure for role modeling is composed of four types of components: And-nodes, Or-nodes, leaf nodes and relations among these nodes. Fig.2 shows our representation for two events in the scene in Fig.1.

An And-node represents an entity that is defined by composing its child nodes, such as a role composed by events, an event by sub-events and, a sub-event by atomic actions. An Or-node connects all variants of an entity. For example, in Fig.2, the Or-node for "events" is connected to all 175 types of different events. Each entity type could be either an And-node which can be further decomposed, or a leaf node which can be easily observed from input data.

The graph defined above is the And-or Graph in [16]. To represent videos, we added an repetitive edge type to represent the fact that a node is composed by multiple repetitions of child nodes, while each child node refers to the same entity happens at different time. For example, the role node in Fig.2 is composed of K events done across the whole video sequence. The dependency edges here are also different because they represent the temporal (instead of spatial) dependency among entities sharing the same immediate parent. We use dashed line to represent the temporal dependency because they are not deterministic relationships. For example, after entering the door, an agent may go to the reception desk, or may go to the time puncher. The last two states must appear after the agent enters the scene, and the direction where agent goes is probabilistic, depending on the role and intent of the agent. The And-or structure defines the space of all possible explanations of agents' trajectories in video. An instance of explanation is realized by specifying for each or-node the active child node it represents, and

instantiating all the leaf nodes. The graph colored in red denotes such a realization, and following [16], is called parse graph.

Formally, the role model of an agent is defined by a 5- tuple: $(r, K, (e_i, pg_i, tr_i)_{i=1}^K)$, where r is the role of the agent, K stands for the number of events (trajectories) of the agent as he/she might be seen multiple times, $e_i \in \{1, 2, ..., 175\}$ stands for the ID of an event, tr_i is a trajectory, and $pg_i = (\{s_i^j\}_{j=1}^3, \{a_i^t\}_{t=1}^{\tau})$ represents for the parse graph of event e_i. Here s_i^j is the ID of the j-th sub-event, a_i^t is atomic action at frame t and τ is the timespan of the corresponding trajectory. For simplify notifications, we define the following collective variables $\mathbf{e} = \{e_i\}_{i=1}^K$ and $\mathbf{pg} = \{pg_i\}_{i=i}^K$. We also denote this trajectory set as $\mathbf{tr} = \{tr_i\}_{i=1}^K$. In this paper, the identification of each agent (at multiple appearances) is manually given[1], which is the only manual input required for using our method to recognize social roles.

Since the model is tree structured, its leaf nodes (i.e. atomic actions) can be derived through an event generating process (i.e. the top-down process of Fig.2): 1) the total number of events performed by the agent is first generated, following a Poisson distribution; 2) the number of events in each event category is decided by a multinomial distribution, where the order of these events does not matter. This is also sometimes called a "loose" grammar because there is no "tight" connections between the nodes; 3) For each event e_i, all possible parse graphs pg_i can be generated by sampling from the And-or structure; 4) As discussed from the previous section, a parse graph is composed of a set of atomic actions.

Therefore, the joint probability of this grammar model can be factorized as the following:

$$p(r, K, \mathbf{e}, \mathbf{pg}, \mathbf{tr}) = p(r)p(K|r)p(\mathbf{e}|K, r)p(\mathbf{pg}, \mathbf{tr}|\mathbf{e}, K), \tag{2}$$

The first RHS term of Eqn.(2) is the prior probability of roles $p(r)$, which is set to uniform distribution in this paper. The second RHS term is the conditional distribution of the number of happened events K given a role category, which we use Possion distribution:

$$p((K|r) = Pois(K, \lambda_r) \tag{3}$$

where λ_r is the parameter for Possion distribution. The reason is that different roles sometimes differs a lot in terns of the number of appearance, which make the Possion distribution a useful tool to distinguish these categories. For example, as illustrated in Fig.4, the "manager" and "visitor" are quite similar in terms of the frequency of events. However, their number of appearance within a unit time differ quite a lot.

The third RHS term of Eqn.(2) is the conditional distribution for the frequency of events given the role category r and the number of instance K, we assume that

$$p(\mathbf{e}|K, r) = Multi(n_1, n_2, ..., n_N; \rho_r) \tag{4}$$

is a multinomial distribution that decides the frequency of each type of events given a role, where $\rho_r = (\rho_r^1, \rho_r^2, ..., \rho_r^N)$ is the parameter vector of multinomial distribution, N is the number of event categories and n_i is the number of events in the i-th event category.

The fourth RHS term of Eqn.(2) can be further decomposed:

$$\begin{aligned} p(\mathbf{pg}, \mathbf{tr}|\mathbf{e}, K) &= \prod_{i=1}^K p(pg_i|e_i)p(tr_i|pg_i) \\ &= \prod_{i=1}^K \prod_{j=1}^3 p(\{s_i^i\}_{j=1}^3|e_i)p(s_i^2|s_i^1)p(s_i^3|s_i^2) \\ &\quad p(\{a_i^t\}_{t=1}^{\tau_i}|s_i^1, s_i^2, s_i^3)p(tr_i^t|a_i^t) \end{aligned} \tag{5}$$

Although there are many terms, this probability function is easy to evaluate since the terms $p(s_i^j|e_i)$ and $p(\{a_i^t\}_{t=1}^{\tau_i}|s_i^1, s_i^2, s_i^3)$ are constants if the decomposition is valid. By valid, we mean those sub-events and atomic actions are child nodes of a possible parse graph in our And-or representation. If the decomposition is invalid, e.g when the sub-event of using a water dispenser have a child atomic action of sitting on the coach, the corresponding probability term would be zero.

For the two temporal constraints $p(s_i^2|s_i^1)$ and $p(s_i^3|s_i^2)$ we directly use non-parametric models that remember the occurrence over any pair of sub-events in the training data. The data term $p(tr_i^t|a_i^t)$ is the Gaussian atomic action likelihood defined as Eqn.(1) in Section 2.2.

Given Eqn.(2), we now define the posterior probability. For an observed set of trajectories from one agent $(K; tr)$, the posterior for role is defined as:

$$\begin{aligned} p(r|K, \mathbf{tr}) &\propto \sum_{\mathbf{e}, \mathbf{pg}} p(K, \mathbf{e}, \mathbf{pg}, \mathbf{tr}|r)p(r) \\ &= p(K|r)E_{p(\mathbf{e}|K,r)}[\sum_{\mathbf{pg}} p(\mathbf{pg}, \mathbf{tr}|\mathbf{e}, K)] \\ &\approx p(K|r)\frac{1}{M}\sum_{m=1}^M \sum_{\mathbf{pg}} p(\mathbf{pg}, \mathbf{tr}|e_m, K) \end{aligned} \tag{6}$$

where $e_m \sim p(\mathbf{e}|K, r)$ is a fair sample from the multinomial distribution given by Eqn.(4).

This equation indicates that the posterior probability of the role can be evaluated by two steps: 1) sampling events $\mathbf{e}$

[1] Ideally, person identification could be done automatically with face recognition methods. However, recognizing person under varying view points and lighting conditions is still a challenging task in itself and has little to do with the rest of this paper.

according to a multinomial distribution; 2) for each sample e_m, computes the summation of joint probability Eqn.(5) with respect to all possible parse graphs.

It is worth mentioning that the model parameters for relation models and the event generating process are fitted by **MLE**, using clustered hotspot positions and role labels of training trajectories. Since we also know the sub-event labels from the previous subsection, we can compute the probability table on sub-event dependencies by pooling over all consecutive sub-event pairs that appear in training data.

4. Grammar Parsing and Bayesian Inference

The inference algorithm of this paper can be divided into two steps.

i) **Parsing trajectories.** Given a trajectories, the task of parsing is to find the parse graph that maximize likelihood of $pg^* = argmax_{pg} p(tr|pg)$. For this purpose, we implemented an online parsing algorithm for And-or graph based on Earleys [7] parser to generate parse graphs based on the input trajectories. Earleys algorithm reads terminal symbols sequentially, creating a set of all pending derivations (sub-events and events) that is consistent with the input up to the current input terminal symbol. Upon reading the next input, this parser iteratively performs one of three basic operations (prediction, scanning and completion) for each state in the current state set. After all the symbols are read, virtually all possible parse graphs for the current trajectory have been evaluated and their probabilities computed. Therefore, the parsing algorithm not only produce the best parse graph pg^* but can also be used to compute the probability summation of all possible parse graphs in Eqn.(6).

ii) **inferring social role.** We repeat the first step for all the observed trajectories, after that we can evaluate each roles posterior function Eqn.(6) by imputing derived valued of Eqn.(5). Therefore, the role of an agent becomes the r that leads to a MAP problem $r^* = argmax_r p(r|K, \mathbf{tr})$.

5. Experiment

5.1. Dataset

To our knowledge, there are not any public benchmark and dataset for conducting social role inference experiments. So, to evaluate our algorithm's performance, we collected a new dataset, which is shot by an wide angle camera in the lobby of a private research lab. The video, which last 8 hours 30 minutes, contains 78 agents which came from 7 role categories and displayed 604 trajectories.

We run a commercial pedestrian tracking algorithm to extract the trajectory bounding box of each person in the dataset. The tracking algorithm successfully tracked all of the single person cases, most of the two and three person cases, and occasionally failed when the scene become really crowded. Since these failure cases only occupy a small portion of cases, which does not affect the results much. As discussed in Section 3, we manually give the trajectories of each person a unique ID. For training and evaluation purpose, the social role of each person is also annotated(which is not required to recognize the social role of a previously unseen person within the same scene).

	Recognition rate
Sub-event: Enter	98.62%
Sub-event: Exit	97.17%
Sub-event: Activity	81.54%
Event	90.28%
Role	87.18%

Table 2. Overall recognition rate for sub-event, event and role.

To evaluate the recognition performance of our algorithm, we also annotated ground truth about events and sub-events on each trajectory. Note that these event and sub-event annotations are not used in training data. In the following experiments, we randomly choose 80% of the agents as training data, the rest as testing data. Experiment results below are the average result of 5 independent runs.

5.2. Parsing Trajectories into Events

For each testing trajectory, we run the modified Earley parsing algorithm specified above, and take the parse graphs. We define the result sub-events and event for each trajectory as sub-events and event in their parse graphs with the highest probability. As shown in the first three rows of Table.2, we can see that the recognition rates of sub-events enter and exit are good, but the recognition rate of activity sub-events is much lower. This is because the relations for atomic actions in activity sub-events are much more complex. However, we can see that the recognition rate of events is higher than that of the activity, this indicates that the temporal dependency between sub-events improves the performance of the model.

5.3. Recognizing Roles of agents

The overall role recognition rate is shown in the last row of Table.2. Considering the fact that the scene is real and reasonably complex, an average recognition rate of 87.2% is already very competitive.

To further illustrate the recognition result, we plotted the confusion matrix for roles. As is shown in Fig.3, we find that the manager and visitor are more prone to be confused with other categories. After analyzing the video, we found that visitors usually go with other roles, so that they are more likely to be confused. The same scenario also happens on managers, as they need to coordinate and collaborate with other roles. This difficulty could also been seen from Fig.4, where we cannot see very clear patterns for visitors and managers.

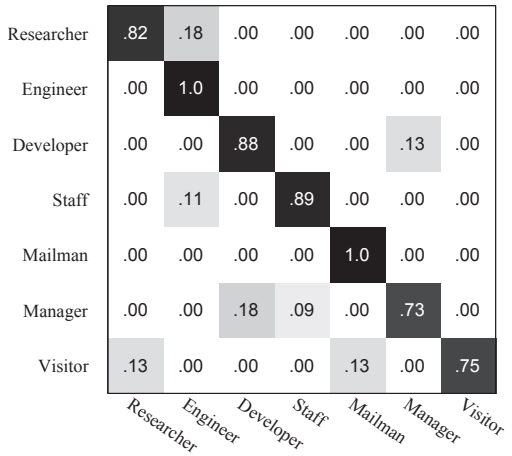

Figure 3. Confusion matrix for roles in the lobby scene. Our algorithm achieves an average role recognition rate of 87.2% on this scene. Difficulty is recognizing Manager and visitor lies in the fact that they usually perform same activity with other agents at the same time. This is also partially evidenced by sampled trajectory plots in Fig.1 and histograms in Fig.4.

We also analyze how the role inference accuracy increases, as more and more trajectories of the agent have been seen. In Fig.5, we take two agents as examples, and show the normalized likelihood of each role. Horizontal axis are the number of events, depth axis are different roles, and height of the bars represent the posterior probabilities of roles at that time. To compute this probability, we replace the parameter λ_r in Eqn.(3) to $\lambda'_r = \frac{\lambda_r}{\tau}t$ where τ is the time span of the full video, and t is the time spot when the currently observed trajectory ends. From the figure, we can see that when agents appear for the first time, the overall ambiguity of our algorithm is high. This is because for the two roles, the first event are not unique to their role (e.g. coming to lab in the morning), so there are much ambiguities among these roles, but as the number of events increase, when the deterministic event happens or the distribution of events is different from others, much of the ambiguities are disappeared. Overall recognition rate of roles and average entropy of the role ambiguity for agents are shown in Fig.6, which again verified this trend and demonstrates the advantage of using long time video for role inference.

6. Discussion

In this paper, we proposed an And-or hierarchical representation and an algorithm for the purpose of inferring roles from long timespan surveillance video. Using our representation, a large number of possible events can be efficiently represented by a relatively small number of atomic actions, which saves time for detecting duplicate atomic actions for different events. With the explicit grammar rules, the Earley parser is used to compute parse graphs for input trajectories, which is further used by bayesian inference to predict the role of agent. Experiment results show that our model can predict the roles of agents with satisfactory recognition accuracy.

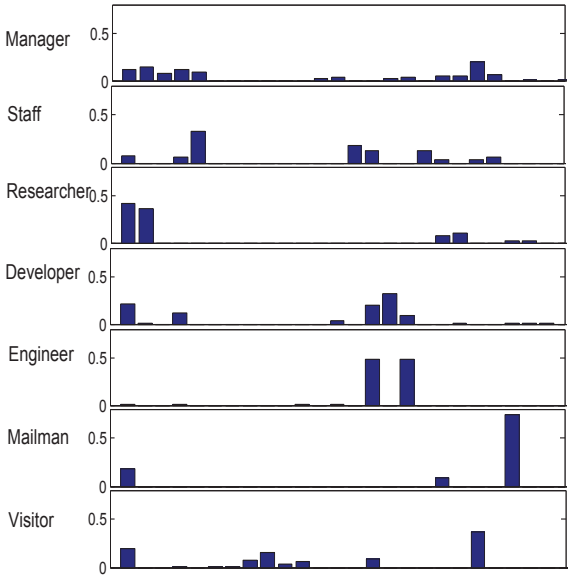

Figure 4. Frequency of events done by each role category. Due to space constraints, we only show the frequency on observed events. From this figure, we can see that although many roles perform the same set of events, their roles can still be differentiated by their frequencies. This implies the necessity of inferring roles from long timespan video.

One limitation of our current method is its limited generability. Given a new surveillance scene, we need re-train the model, which entails annotating social role of many agents. This requirement might be impractical for real-world applications. However, since our model is based on interactions between agents and semantic hotspots, it is possible to perform transfer learning by simply specifying the correspondence between hotspots in the old scene and those in the new scene. Intuitively, this method should be effective for places with fixed hotspot configurations like fast-food franchise, hotel lobby and even bank lobby. This is the future direction of our research.

Acknowledgement. The work was supported by the Innovation Team Development Program of the Chinese Ministry of Education Grant No.:IRT0606 and the National Natural Science Foundation of China Grant No.60827003, No.60832004.

References

[1] 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. 2

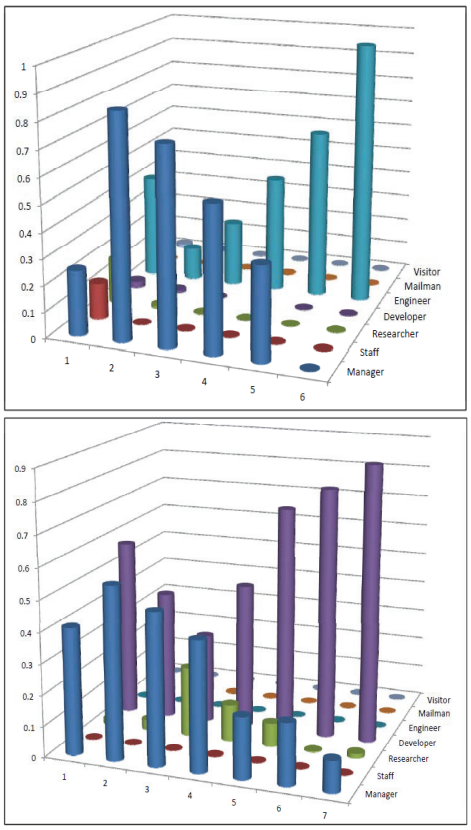

Figure 5. Role probability change as the number of observed trajectory increases.The top and bottom are two randomly chosen agents from the scene. Where horizontal axis are the number of events, depth axis are different roles, and height of the bars represent the posterior probabilities of roles at that time.

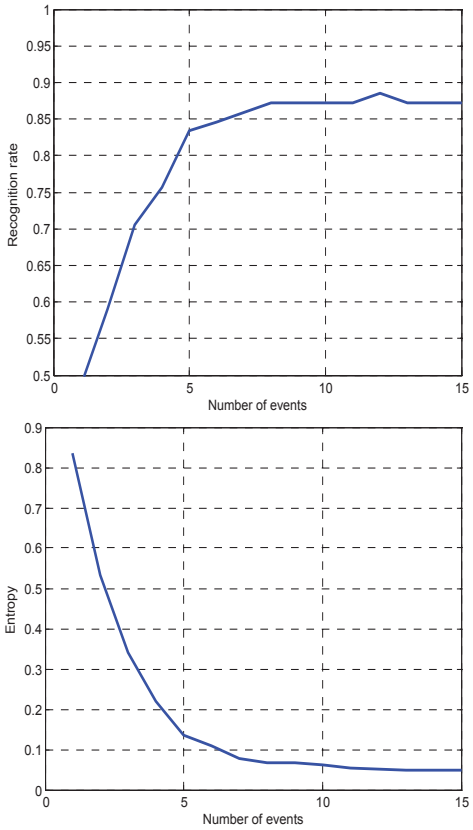

Figure 6. Overall role recognition rate(top) and entropy of role posterior probability(bottom), as a function of the number of observed trajectories for each agent. We can see that as more and more trajectories are observed, the role ambiguity decreases, and role recognition rate increases. This gain demonstrates the advantage of using long timespan videos for role inference.

[2] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. *CVPR*, 1997. 2

[3] L.-B. Chang, Y. Jin, W. Zhang, and E. B. S. Geman. Context, computation, and optimal roc performance in hierarchical models. *IJCV*, 2010. 2

[4] R. Dechter and R. Mateescu. AND/OR search spaces for graphical models. *Artificial Intelligence*, 171(2-3):73–106, 2007. 2

[5] L. Ding and A. Yilmaz. Learning relations among movie characters: a social network perspective. In *ECCV*, 2010. 1

[6] J. K. Hamlin, K. Wynn, and P. Bloom. Social evaluation by preverbal infants. *Nature*, 450(7169):557–559, Nov. 2007. 1

[7] J.C.Earley. *An Efficient Context-Free Parsing Algorithm*. PhD thesis, Carnegie-Mellon Univ, 1968. 2, 6

[8] M.S.Ryoo and J.K.Aggarwal. Recognition of composite human activities through context-free grammar based representation. *CVPR*, pages 1709–1718, 2006. 2

[9] P. Natarajan and R. Nevatia. Coupled hidden semi markov models for activity recognition. *IEEE Workshop on Motion and Video Computing*, 2007. 2

[10] J. Pearl. *Heuristics: intelligent search strategies for computer problem solving*. Addison-Wesley Long-man Publishing Co., Inc, Boston, MA, 1984. 2

[11] M. Pei, Y. Jia, and S.-C. Zhu. Parsing video events with goal inference and intent prediction. 2011. 2, 3

[12] A. Stolcke. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 1995. 2

[13] T. D. Ullman, C. L. Baker, M. Owen, E. Owain, N. D. Goodman, and J. B. Tenenbaum. Help or hinder : Bayesian models of social goal inference. *NIPS*, 2009. 1

[14] T. Yu, S.-N. Lim, K. Patwardhan, and N. Krahnstoever. Monitoring, recognizing and discovering social networks. *CVPR*, 2009. 1

[15] Z. Zhang, T. Tan, and K. Huang. An extended grammar system for learning and recognizing complex visual events. *PAMI*, 2011. 2

[16] S.-C. Zhu and D. Mumford. A stochastic grammar of images. *Foundat. Trends Comput graphics Vision*, 2:259–362, 2007. 2, 4, 5