Event Modeling and Recognition using Markov Logic Networks *

Son D. Tran and Larry S. Davis

Department of Computer Science
University of Maryland, College Park, MD 20742 USA
{sontran, lsd}@cs.umd.edu

Abstract. We address the problem of visual event recognition in surveillance where noise and missing observations are serious problems. Common sense domain knowledge is exploited to overcome them. The knowledge is represented as first-order logic production rules with associated weights to indicate their confidence. These rules are used in combination with a relaxed deduction algorithm to construct a network of grounded atoms, the Markov Logic Network. The network is used to perform probabilistic inference for input queries about events of interest. The system's performance is demonstrated on a number of videos from a parking lot domain that contains complex interactions of people and vehicles.

1 Introduction

We consider the problem of event modelling and recognition in visual surveillance and introduce an approach based on Markov Logic Networks ([1]) that naturally integrates common sense reasoning with uncertain analyses produced by computer vision algorithms for object detection, tracking and movement recognition. We motivate and illustrate our approach in the context of monitoring a parking lot, with the goal of matching people to the vehicles they arrive and depart in.

There are numerous frameworks for event recognition. In declarative approaches (e.g. [2]), events are represented with declarative templates. Events are typically organized in a hierarchy, starting with primitive events at the bottom and composite events on top. The recognition of a composite event proceeds in a bottom-up manner. These approaches have several drawbacks. First, a miss or false detection of a primitive event, which occurs frequently in computer vision, especially in crowded or poorly illuminated conditions, often leads to irrecoverable failures in composite event recognition. Second, uncertainty is often not modelled and so these methods are generally not robust to typical errors in image analysis. In probabilistic frameworks, such as HMMs (e.g. [3]), or DBNs ([2]), events are represented with probabilistic models. Event recognition is usually performed using maximum likelihood estimation given observation sequences. While these approaches provide robustness to uncertainty in image analysis, their representations often lack flexibility (e.g. number of states or actors is fixed) and hence it is difficult to use them in dynamic situations.

* This research was funded in part by the U.S. Government VACE program

In general, the problems of noise or missing observations always exist in real world applications. Our contention is that common sense knowledge, specific to the domain under consideration, can provide useful constraints to reduce uncertainties and ambiguities. Having a good knowledge base (KB) and an effective reasoning scheme helps to improve event recognition performance.

Technically, we address uncertainty in observations and representational richness of event specification by a combination of logical and probabilistic models.

1. Domain common sense knowledge is represented using first order logic statements. Both negation and disjunction are allowed.
2. Uncertainty of primitive event detection is represented using detection probabilities. Uncertainty of logical relations (including event models or logical constraints) is represented with a real-valued weight set based on, for example, domain knowledge.
3. Logical statements and probabilities are combined into a single framework using Markov Logic Networks (MLN, [1]).

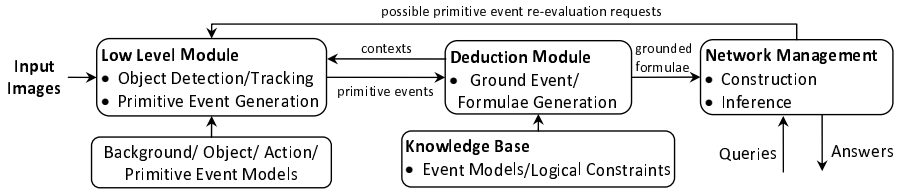

Fig. 1. Overview of our system

Our system maintains an undirected network of grounded atoms which correspond to events that have occurred in the video. At any moment, primitive events are detected with associated detection probabilities. They are then used to ground logical rules in the KB, which generally leads to generating more grounded events. Next, these grounded logical rules are added to the Markov network. The network parameters or structures are revised with these updates. The marginal probability of any (composite) event can be determined using probabilistic inference on this network. Fig. 1 shows an overview of our system.

2 Related Work

Visual event detection from video has a long history in computer vision. We review here only approaches that are most relevant to ours. Logic has been used for visual event recognition in a number of works. In [4], Rota et al presented an elegant treatment for representing activities using declarative models. Recognition was performed effectively using a constraint-satisfaction algorithm. In [5], Shet et al used a multi-valued default logic for the problem of identity maintenance. Default reasoning was conducted on a bi-lattice of truth values with prioritized default rules. Identity maintenance rules are prioritized mainly based on domain

knowledge. A continuous bi-lattice was used in [6] for human detection. Here, instead of using a multi-valued logic, we use a combination of logic and probability to handle inexact inference including identity maintenance. Each level of prioritization can be mapped to our framework using a rule weight.

The combination of probability and (first order) logic has been pursued extensively in AI and led to the emergence of Statistical Relation Learning (SRL, [7]). SRL representations often involve unrolled or grounded graphical models (directed (Bayesian) as well as undirected (Markov) ones), which are constructed using a frame-based or a logic-based approach. They have been used for human activity recognition (although not in vision-based systems). In [8], Liao et al recognized human activities based on the information about locations they visited provided by GPS sensors worn by users (location-based). Probabilistic inference is performed on an unrolled Markov network formed from a Relational Markov Network, which essentially encodes high-level domain knowledge. Relation weights can be learned using a MAP estimation technique. In [9], Pentney et al recognized human activities based on objects used. The objects were RFID tagged and identified using RFID readers worn by the users (object-use based). Logical rules are grounded and linked to form a probabilistic network within a single time slice. In general, these approaches are intrusive. They require users to wear additional sensors. Therefore their application to general surveillance tasks is limited. Here, we work with visual input and use common sense knowledge to complement limitations in visual perception. Markov Logic Networks were used to construct a DBN for activity recognition by Biswas et al in [10]. That work only addressed inaccuracy in logic statements, while ours addresses a wider range of issues, including detection uncertainty, missing observations and identity maintenance.

Approaches that are based on probabilistic grammars for event recognition such as [11] typically use simpler rules than ours. For example, they do not allow existential quantifiers, which are needed for dealing with missing observations (Section 5.2). It is also difficult to express domain constraints such as "a car can only be driven by one person" using generative grammars. Furthermore, methods to perform probabilistic propagation are better understood for graphical models than for probabilistic grammars.

3 Sample Problem

We motivate our approach with the surveillance problem of monitoring a parking lot and determining which people enter or leave in which cars (Fig. 2). In a parking lot, cars of various shapes and sizes can park close together. Occlusion is not only unavoidable but sometime severe. This leads to many difficulties in tracking people, since their corresponding foreground blobs may change from complete to fragmented or become totally missing as they move between parked cars. It is often difficult to determine the exact moment a person enters a car, or even which car a person enters. Pure declarative, bottom-up approaches (e.g. [2], [4]) that rely on accurate capture of primitive events will not work well here. Probabilistic action recognition (e.g. HMMs([3])) might fail as well since, locally, an observation may be missing altogether. For more robust event recognition, we

Fig. 2. A frame from a parking lot sequence and its corresponding foreground regions detected using background subtraction. Here, parked cars introduce significant occlusion and door openings lead to many false alarms.

propose to use common sense knowledge about the domain under consideration. The knowledge base will contain rules that range from definite ones such as "if a car leaves, there must exist some person driving it" or "a person can drive only one car at any time" to weaker rules such as "people walking together usually enter the same car" or "if a person puts a bag into the trunk of a car, he or she is likely to enter that car". We represent these rules using first order logic augmented with probabilities to represent the degree of confidence for each rule. Additionally, the recognition of actions or primitive events such as "walking together" or "put a bag into a car" is uncertain. Probability theory provides a convenient method to handle these also. We then need an approach that combines logic and probabilistic elements in a coherent framework.

Briefly, we achieve this by using 1) first order logic formulae to represent domain knowledge, 2) a real-valued weight to represent the confidence in each logic rule, 3) probability to model uncertainty for primitive event and action recognition, 4) a probabilistic logic network, namely the Markov Logic Network, to connect (detected) ground atoms and to perform probabilistic inference (e.g. determine the probability that a person enters some car, given the input sequences). The following sections discuss in detail these aspects for our particular surveillance problem as well as for general surveillance contexts.

4 Background on Markov Logic Networks

Markov Logic Networks (MLN, [1]) are one type of the unrolled graphical models developed in SRL([7]) to combine logical and probabilistic reasoning. In MLN, every logic formula F_i is associated with a nonnegative real-valued weight w_i. Every instantiation of F_i is given the same weight. An undirected network, called a Markov Network, is constructed such that,

- Each of its nodes correspond to a ground atom x_k.
- If a subset of ground atoms $x_{\{i\}} = \{x_k\}$ are related to each other by a formula F_i, then a clique C_i over these variables is added to the network. C_i is associated with a weight w_i and a feature f_i defined as follows

$$\begin{aligned} f_i(x_{\{i\}}) &= 1, \text{ if } F_i(x_{\{i\}}) \text{ is true,} \\ &= 0, \text{ otherwise .} \end{aligned} \tag{1}$$

Thus first-order logic formulae in our knowledge base serve as templates to construct the Markov Network. This network models the joint distribution of the set of all ground atoms, X, each of which is a binary variable. It provides a means for performing probabilistic inference.

$$P(X = x) = \frac{1}{Z} \exp(\sum_i w_i f_i(x_{\{i\}})). \tag{2}$$

where Z is the normalizing factor, $Z = \sum_{X \in \mathbf{X}} \exp(\sum_i w_i f_i(x_{\{i\}}))$. If $\phi_i(x_{\{i\}})$ is the potential function defined over a clique C_i, then $log(\phi_i(x_{\{i\}})) = w_i f_i(x_{\{i\}})$.

Inference Based on the constructed Markov network, the marginal distribution of any event given some evidence (observations) can be computed using probabilistic inference. Since the structure of the network may be very complex (e.g. containing undirected cycles), exact inference is often intractable. MCMC sampling is a good choice for approximate reasoning ([1]). In MLN, the probability that a ground atom X_i is equal to x_i given its Markov blanket (neighbors) B_i is

$$P(X_i = x_i | B_i = b_i) = \frac{\exp(\sum_{f_j \in F_i} w_j f_j(X_i = x_i, B_i = b_i))}{\exp(\sum_{f_j \in F_i} w_j f_j(X_i = 0, B_i = b_i)) + \exp(\sum_{f_j \in F_i} w_j f_j(X_i = 1, B_i = b_i))} \tag{3}$$

where F_i is the set of all cliques that contain X_i and f_j is computed as in Eq. 1.

Basic MCMC (Gibb sampling) is known to have difficulty dealing with deterministic relations, which are unavoidable in our case. It has been observed that using simulated tempering ([12]) gives better performance than the basic Gibb sampling ([12]). Simulated tempering is a MC method that is closely related to simulated annealing. However, instead of using some fixed cooling schedule, a random walk is also performed in the temperature space whose structure is predetermined and discrete ([12]). These moves aim at making the sampling better at jumping out of local minima.

5 Knowledge Representation

In this section, we will describe our approach to represent knowledge and its associated uncertainty. In our framework, object states and their interactions (including the so-called events, actions or activities as they are interchangeably referred to in other work, e.g. [2], [4]) are all represented with first order logic predicates. A predicate is intensional if its truth value (for a certain grounding of its arguments) can only be inferred (i.e. cannot be directly observed) ([13]). A predicate is extensional if its truth value can be directly evaluated by a low-level vision module. It is strictly extensional if this is the only means to evaluate it (i.e. it can only be observed and not inferred).

5.1 Logical Representation

In [1], the Markov network is constructed using an exhaustive grounding scheme, which can lead to an explosion in the number of ground atoms and network connections. Most of them are irrelevant and create significant difficulties for inference. A more efficient scheme was proposed in [14], which essentially grounded only clauses that can become unsatisfied using a greedy search. It is not clear if this approach could handle dynamic domains that involve, for example, time and location. Here, we represent our knowledge in the form of production rules, $production \rightarrow conclusion$, and use deduction to ground (and add to the Markov network) only literals (including both positive and negative atoms) that are possibly true.

In traditional deductive systems (e.g. [13]), production rules in the form of Horn clauses are used extensively. However, Horn clauses cannot represent negations and disjunctions, which are often required to capture useful commonsense knowledge. To increase our system's representational ability, we allow the following rule forms,

- $(\bigwedge_i a_i) \rightarrow b$ Definite (i.e. Horn) clauses are used to define a composite event from sub-events (similar, for example, to multi-thread event definition in [2]), causal and explanatory relationships between observations and underlying actions (e. g. $use(Bowl) \rightarrow make(Cereal)$ or $at(Resstautrant) \rightarrow have(Dinner)$ in object-use based [9] and location-based frameworks [8])
- $(\bigwedge_i a_i)(\bigwedge_j \neg b_j) \rightarrow c$ Many events can only be described with a rule that has negative preconditions, for example, $at(C,S,t) \wedge \neg stopped(C,t) \rightarrow violate(C,S,t)$ where C is a car and S is a stop sign. Identity maintenance ([5], [15]) also often leads to formulae with negative preconditions, for example, $own(H_1, Bag) \wedge take(H_2, Bag) \wedge \neg eq(H_1, H_2) \rightarrow theft(H_2, Bag)$.
- $(\bigwedge_i a_i) \rightarrow \neg b$ This form is often used to describe an exclusion relation. For example, the rule "a person P belongs to only one group G" can be written as $belongto(G_1, P) \wedge \neg eq(G_1, G_2) \rightarrow \neg belongto(G_2, P)$.
- $(\bigwedge_i a_i) \rightarrow (\bigvee_j b_j)$ Disjunctions are used when a single conclusion cannot be made. For example, $use(Cup) \rightarrow (drink(Coffee) \vee drink(Tea))$. When it fires, all atoms in the conclusion are added to the ground atom database. Disjunctions also arise from existential quantifiers (next section).

These forms, of course, are not the most general ones in First Order Logic. However, practically, they are sufficiently rich to represent a wide range of common sense knowledge and to capture complex events in surveillance domains.

5.2 Uncertainty Representation

Uncertainty is unavoidable in practical visual surveillance applications. We consider two classes of uncertainty: logical ambiguity and detection uncertainty. Their sources and ways to represent them are described below.

Incomplete or Missing Observations Occlusion and bad imaging conditions (e.g. dark, shadowed areas of the scene) are two common conditions that prevent us from observing the occurrence of some actions. In some cases, even if a unique conclusion cannot be made, some weaker (disjunctional) assertion might still be possible. Rules with disjunctive effects are often needed then. For example, the statement "if a bag b is missing at some time interval t and location L, then someone must have picked it up" could be formalized as $missing(b, l, t) \rightarrow (\exists p\ passBy(p, l, t) \wedge pickUp(p, b, t))$. Here the action $pickUp(p, b, t)$ can be inferred when its direct detection is missed. This type of formulae involves an existential quantifier and will be expanded to a disjunction of conjunctive clauses when grounded. For example, suppose that $passBy(P_1, L, T)$ and $passBy(P_2, L, T)$ are true for two persons P_1 and P_2 (i.e. two persons P_1 and P_2 passed by when the bag went missing), then the grounding of this rule would be $missing(B, L, T) \rightarrow (passBy(P_1, L, T) \wedge pickUp(P_2, B, T)) \vee (passBy(P_2, L, T) \wedge pickUp(P_2, B, T))$. This expansion obviously is not suitable for infinite domains. However, in practice, most object domains are finite (e.g. number of people or cars is finite) therefore the expansion is feasible for surveillance. As evidence arrives, previously expanded domains may need to be updated (section 6.1).

Non-perfect Logical Statements Common sense statements in the KB are not always true. We use a real-value weight to represent the confidence of each rule in the KB. Rules with absolute certainty, such as "a person can drive only one car at a time", are given an infinite weight. In practice, such a hard clause is "softened" with a maximum weight, $MAXW$, to facilitate the inference process. Rules that are almost always true, such as "a person interacts with only one car", are given strong weights. Weak weights are assigned to rules that describe exceptions (i.e. situations that are possibly true but not common such as "a driver might enter a car from the passenger side").

Extensional Evaluation Uncertainty The evaluation of an extensional predicate, E, by the low-level vision module might return answers with absolute certainty or with some associated (detection) probability, $p_D(E = true)$. For the first case, whether the result is $true$ or $false$, we make E an evidence variable and add it to the Markov network. For the second case, a method to integrate E and its detection probability for high-level logical reasoning are needed.

One approach would be to add this grounded, single-atom clause, $(E, w \propto p_D)$ and its complement, $(E, w \propto 1 - p_D)$ to the Markov network. (Note that using only one of these clauses is not sufficient). This way, the marginal probability, $p(E = true)$, is fixed to p_D. However, evidence from other sources may change the probability $p(E = true)$, especially when E is not strictly extensional. Therefore, it would be better to add an observation variable O and use these two formulae: $(observe(O) \rightarrow E, w \propto p_D)$ and $(observe(O) \rightarrow E, w \propto 1 - p_D)$. The variable O has a fixed value that represents the corresponding measurement. It is specific to this grounding. The predicate $observe(O)$ will not take part in any logical deduction and is always assumed true. This formulation allows evidence from related sources (beside O) to have their effects on $p(E = true)$.

Extensional predicates can be of various kinds depending on the domain under consideration. Two classes and their associated uncertainty that we consider are object recognition and action detection (see section 7.1).

Identity Maintenance Identity maintenance is necessary when there exist multiple identities that actually refer to the same object([5], [15]). In surveillance, it is caused by lack of visual information (appearance, shape...) to make unique identity connections across observation gaps. Our approach to solve this problem is similar to the one proposed in [15] for entity resolution in relational databases ([7]), with a slightly more concise formulation.

Identification of two objects A and B is represented by a predicate $eq(A, B)$. It comes with the following set of axioms (with infinite weights): 1) *Reflexive*, $eq(A, A)$; 2) *Symmetry*, $eq(A, B) \leftrightarrow eq(B, A)$; 3) *Transitivity*, $eq(A, B) \wedge eq(B, C) \rightarrow eq(A, C)$; 4) *Predicate Equivalence*, $P(X_1, Y) \wedge eq(X_1, X_2) \rightarrow P(X_2, Y)$, (for two-ary predicates but can be similarly stated for n-ary predicates).

The equivalence predicate can be extensionally evaluated or intensionally inferred. Extensional evaluation of $eq(A, B)$ is done using appearance matching. The probability $p(eq(A, B) = true)$ is calculated based on a matching score. Intensional deduction of $eq(A, B)$ can be done using the above axioms and commonsense rules in the KB. Several prioritized rules in [5], such as "possession of some special objects (e.g. car keys) determines owners' identity", can be used here, where each prioritizing level is mapped to a corresponding weight.

6 Network Construction

This section describes our deduction algorithm that uses the production rules in the KB (section 5.1) to deduce grounded atoms for the Markov network. Due to noise or incompleteness in observations, some events that have not actually occurred might get grounded and added to the ground atom database (ADB). Our procedure is thus a relaxed version of logical deduction and may not be logically consistent.

6.1 Deduction Algorithm

Typically, with definite clauses, deduction is performed via forward chaining. In our system, logic rules take richer forms that require us to additionally deal with negative preconditions and disjunctive conclusions. Following are several preliminaries for our algorithm.

Close World Assumption(CWA) Since it is usually not convenient and sometimes impossible to detect (consistently) events that are not happening, such as the $notstopped(C_i, t)$ event (for all cars at all time points), the CWA is used to check for negative preconditions: what is not currently known to be true is assumed false. Then, forward chaining is still used, but is divided into two phases: the first for rules that do not have negative preconditions and the second for the remaining rules. This is to delay, for example, the conclusion that $\neg a$ is true using the CWA until all possible ways of deducing a have been tried.

Context-dependent Preconditions Consider the predicate $nearBy(P, loc, t)$ in the formula $happenAt(E, loc, t) \wedge nearBy(P, loc, t) \rightarrow witness(E, P)$. It would be cumbersome to evaluate $nearBy(P, loc, t)$ and add it to the ADB for all people P, all locations loc and all times t. Instead, it should only be evaluated after $happenAt(E, loc, t)$ is true with specific bindings of loc and t. In this case, the satisfaction of the first precondition serves as the context that enables the lazy evaluation of the second one. Generally, we use lazy evaluation for an extensional predicate when it would be expensive to evaluate otherwise due to the large size of the domain (e.g. ones that involve time or location).

Disjunction Domains Generally, in our system, disjunctions need no special treatment. However, when they are in the scope of an existential quantifier, domain expansion and several bookkeeping steps are required. In the missing bag example in section 5.2, the predicate $passBy(P, L, T)$ limits P to the set $\{P_1, P_2\}$ and the existential quantifier is expanded over the entirety of this domain. In general, we eliminate the existential quantifier by considering that the conclusion has two parts, one for defining the object domain ($passBy(P, L, T)$) and the other for describing the actual conclusion ($pickUp(P, B, T)$) In other words, our general production rule would be $precondition \rightarrow (\exists x\ domaindef_x \wedge conlusion_x)$. An empty clause $domaindef_x$ implies that the domain consists of all instantiations of x. During deduction, we may need to expand domains as new objects that satisfy domain predicates are discovered. In such cases, the previously grounded formula is replaced with the new one and the network is modified with the new clique.

Ground Atom and Formulae Deduction

- *Input* ADB - ground atom database; KB_{pos} - set of definite rules; KB_{neg} - set of rules that have negative preconditions
- *Output* ADB - with new ground atoms added; GS - the set of grounded clauses.

Repeat until no new ground atom is generated

1. Repeat until no new atom
   For $\forall R \in KB_{pos}$, instantiate R w. r. t. ADB and for each instantiation r,
   (a) If all context-independent preconditions are satisfied, then evaluate all context-dependent preconditions and add the newly evaluated atoms to ADB.
   (b) If all succeeded, get effects and add to ADB.
   (c) $GS \leftarrow GS \cup r$.
2. Similar to step 1 for $R \in KB_{neg}$ with CWA added during instantiation of rules.

Fig. 3. The algorithm for deducing new ground atoms

The deduction procedure is shown in Fig. 3. In step 1(a), when grounding a clause, if context-independent preconditions are satisfied then context-dependent predicates will be extensionally evaluated. Instances that are evaluated to true will be added to the ADB. In step 1(b), all atoms in the conclusion as well

as their complemented literals (i.e. E and $\neg E$) are added to the ADB. If an existential quantifier is involved, we need to check and update, if necessary, its previously expanded domain. Step 2 essentially repeats step 1 with the addition of the CWA. For a precondition, $\neg E$, if we are unable to observe or deduce E, $\neg E$ is assumed true. If the related clause ends up being grounded (i.e. all other preconditions are evaluated to true) then the literal $\neg E$ is added to the ADB.

All ground clauses are then added to the Markov network. This construction procedure is performed whenever there is a new event generated. It can be done incrementally by deriving only deductions that originate from new events.

7 Implementation and Experiments

7.1 Implementation

We describe here some basic elements needed to address the parking lot application: object set, predicate set, their evaluation and the KB. Three types of objects are considered: cars (denoted as C_i), humans (H_i) and locations (L_i). Time is represented using atomic intervals with granularity of n_I frames (e.g. $n_I = 30$, approximately 2 seconds). Each primitive event or action is assumed to be true within one time interval. Below, time labels are omitted for clarity. Our vocabulary consists of the following predicates: extensional, context-independent, $inTrunkZone(C, H)$, $inLeftZone(C, H)$, $inRightZone(C, H)$, $disappear(H, L)$, $equal(H_1, H_2)$, $shakeHand(H_1, H_2)$ and $carLeave(C)$; extensional, context-dependent, $openTrunk(C, H)$; intensional: $enter(C, H)$ and $drive(C, H)$. Additionally, we have measurement objects and their corresponding predicates (Sec. 5.2).

Background subtraction, human detection and tracking (see e.g. [16]) techniques were first applied to identify and track object locations. The orientation and direction of each car were estimated simply using its corresponding foreground blob and parking lot layout. Fig. 4.1 shows the estimated layouts of the three detected cars during one experiment.

A spatial predicate, for example, $inTrunkZone(C, H)$, is generated when the foot location of person, H, intersects significantly with the trunk zone of the car, C, for a sufficiently long period of time; $disappear(H, L)$ is generated when we lose track of H. Identity maintenance predicates are evaluated using the distance between color histograms of the two participating objects. $shakeHand(H_1, H_2)$ is modeled by analyzing the connecting area between two standing separate persons. $openTrunk(C, H)$ is evaluated base on the motion pattern in the trunk area of car C. The rules that constitute our knowledgebase are listed in the appendix. The maximum weight, $MAXW$, is set to be proportional to the network's size (number of ground atoms [12]). The range $0 - MAXW$ is uniformly discretized to five levels corresponding to very strong, strong, medium, weak and very weak certainties. These values are assigned to rules according to our confidence in them, based on domain knowledge.

7.2 Experiments

We analyzed a set of parking lot videos that involve a number of people entering different cars as listed in Table 1. A typical scenario is as follows. Initially, three

Table 1. Four sequences used in our experiments

	♯ of people	♯ of cars	Durations
seq 1	6	3	2 min 10 sec
seq 2	5	3	3 min
seq 3	4	2	1min 30 sec
seq 4	6	3	4 min

cars, C_1, C_2 and C_3, park next to each other. A person H_1 appears, walks up to C_2, opens its trunk (Fig. 4.2), puts something in, closes the trunk and then disappears between C_1 and C_2 (Fig. 4.3). Then two persons, H_2 and H_3, walk close to each other near the parked cars. They shake hands (Fig. 4.4) and disappear between C_2, C_3 and around the left of C_3 respectively (Fig. 4.5). Person H_4 walks to C_1 and disappears from the passenger side of C_1 (Fig. 4.5). A person H_5 follows a similar path (Fig. 4.6). Person H_6 walks to the cars and disappears between C_2 and C_3 (Fig. 4.7). Then C_1 pulls out and leaves. Finally, C_2 and C_3 follow in order (Fig. 4.8). This scenario consists of a number of interesting interactions and we would like to query the system about who entered or drove which car. The challenges arise mostly due to noise and occlusion, which lead to loss of track well before a person enters a car.

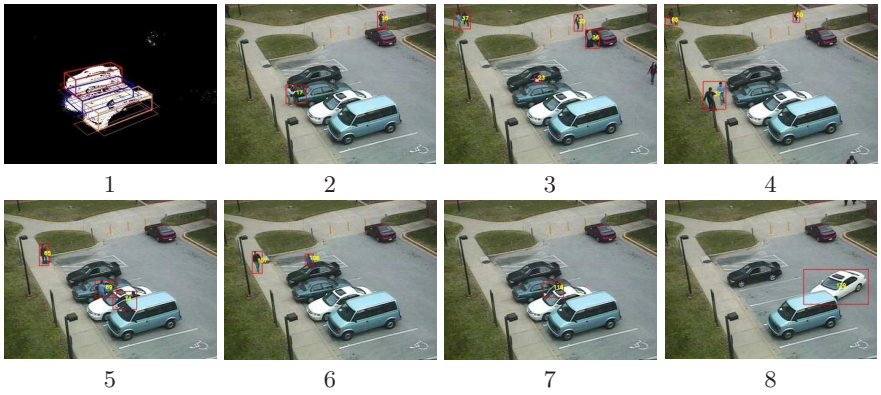

Fig. 4. 1) Estimated layouts of the parked cars. 2-8) Key frames from the scenario. Foreground and human detection results are also shown. Irrelevant people and cars are removed.

As the scenario unfolds, new events are generated and our ground network evolves accordingly. We can query our system at any instant of time. Here, we ran queries after all cars had departed. Detection probabilities for $openTrunk(C_2, H_1)$ and $shakeHand(H_2, H_3)$ were respectively 0.9 and 0.5. Identity confusion is not significant so no related ground atom is generated. Fig. 5 shows the results for our queries, the probabilities $p(enter(c, h) = true)$ and

$p(drive(c,h) = true)$ for all cars C_i's and human H_i's. These results can be

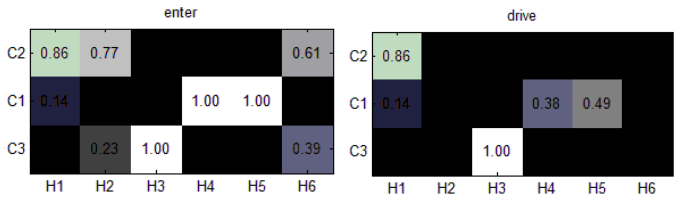

Fig. 5. Right, probabilities of entering cars for individual persons, $p(enter(c,h) = true)$. Left, probabilities of driving cars, $p(drive(c,h) = true)$. Darker means being lower in value

explained intuitively as follows. Person H_1 disappeared between car C_1 and C_2, so he could have equally entered either of them (with $p = 0.5$). But since he was observed to open C_2's trunk, the probability that he entered C_2 increased to 0.86. Person H_2 could have entered either C_2 or C_3 (each with $p = 0.5$). But he had been detected shaking hands with H_3 (and so probably saying "goodbye"), who entered C_3 with high certainty. Hence, the probability of H_2 entering C_3 was reduced and entering C_2 was increased. However, since the detection probability was not very high ($p = 0.5$), the increase was not as much as for the first person. Persons H_3, H_4 and H_5 entered cars C_3, C_1 and C_1 respectively with high certainties since there was only a single car close to them when they disappeared. Person H_6 disappeared between cars C_1 and C_3. Since there is no further supporting evidence, the probabilities for entering C_1 and C_3 should be the same. The observed discrepancy is due mainly to sampling approximation.

H_1 and H_3 drove C_2 and C_3, respectively, with high certainty since they had been observed to enter their cars from the driver side and there was no competing alternative. For car C_1, H_4 and H_5 were observed to enter it from the passenger (right) side. The only person who could possibly enter it from the driver side was H_1. But it was strongly believed H_1 entered and drove C_2. Hence, no one entered C_1 from the driver side. This led to the conclusion that either H_4 or H_5 had moved from the passenger side to drive C_1. This reasoning was made possible because of the low-weight rule 5, which states that, although unlikely, drivers can enter a car from the passenger side (the driver side may be blocked). In general, this type of rule is added to increase the system's ability to handle exceptional cases. They bear some of the flavor of default reasoning in the sense that in the absence of supporting evidence for "normal" rules, rules with low weights will be activated to explain the situation under consideration. Note that rule 5 was also applied to other persons, such as H_2 and H_6, but produced negligible effects since competing hypotheses were much stronger. Therefore, the probabilities that H_2 and H_6 drove any car were still close to zero.

In the initial querying, our system was able to conclude that either H_4 or H_5 drove car C_1 but was unable to determine which of them did. Consider adding to the KB a very weak rule stating that among the persons entering a car from the passenger side, whoever enters it first is its driver (no new ex-

tensional predicate evaluation is needed). When we do this and re-evaluate the queries, $p(drive(C_1, H_4) = true)$ increased to 0.60 while $p(drive(C_1, H_5) = true)$ dropped to 0.23, which matched the observation that H_4 entered C_1 before H_5 and drove it away. This example is intended to illustrate that domain knowledge can be easily added to our system to improve its performance. This would not be so easy in a purely logical system, because a significant amount of consistency checking is required before admitting a new piece of knowledge. Also, it would not be straightforward in a purely probabilistic system, since adding complicated structures involves cumbersome changes to the internal structure of the system.

Our system is implemented using C++ and runs in Windows and Cygwin on an Intel Pentium Dual-Core 1.6 GHz machine. The average running time for background subtraction, human detection and object tracking, on 640x480 RGB frames, are 5 frames per second. Average running time when performing inference for all queries presented here is approximately fifteen seconds (maximum number of MCMC step is set to 5000).

8 Discussion

We described how a combination of a probabilistic graphical model, the Markov Logic network, and first-order logic statements can be used for event recognition in surveillance domains, where unobservable events and uncertainties in detection are common. Logic provides a convenient mechanism to utilize domain knowledge to reason about the unobservable. Probabilistic models give us a coherent framework to deal with uncertainty. The combination brings us the ability to capture interesting interactions in complex domains.

Obtaining a sufficient and efficient knowledge base (KB) is important to our recognition performance. This issue is also receiving increasing attention in the area of human activity understanding([9], [8]). Approaches that use efforts from open communities to build common sense KB have been proposed such as Open Common Sense and Open Indoor Initiatives ([17], [18]). Large databases of rules are now publicly available([17]). Automated rule extraction and weight learning from them have been attempted (e.g. [9]). At its current stage, for visual surveillance, rules and facts collected from such open databases are more often than not irrelevant and not useful. For example, a query for "parking lot" would output relations such as it is "a place for parked cars", "would be hot in the summer", "may or may not be empty"... ([17]). However, as these knowledge bases grow and, possibly, specialize ([18]), their application to our framework seems promising. Exploiting them is part of our future investigation.

References

1. Richardson, M., Domingos, P.: Markov Logic Networks. Machine Learning **62** (2006) 107–136
2. Nevatia, R., Zhao, T., Hongeng, S.: Hierarchical language-based representation of events in video stream. In: Proc. of CVPRW on Event Mining, IEEE (2003) 39–47
3. Yamato, J., Ohya, J., Ishi, K.: Recognizing human action in time-sequential images using Hidden Markov models. In: Proc. CVPR92, IEEE (1992) 379–385

4. Rota, N., Thonnat, M.: Activity recognition from video sequences using declarative models. In: Proc. ECAI02. (2002) 673–680
5. Shet, V., Harwood, D., Davis, L.: Multivalued default logic for identity maintenance in visual surveillance. In: Proc. ECCV06. Volume 4 of LNCS., Berlin Heidelberg New York, Springer (2006) 119–132
6. Shet, V., Neumann, J., Ramesh, V., Davis, L.: Bilattice-based logical reasoning for human detection. In: Proc. CVPR07, IEEE (2007) 1–8
7. Getoor, L., Taskar, B.: Intro. to Statistical Relational Learning. MIT Press (2007)
8. Liao, L., Fox, D., Kautz, H.: Location-based activity recognition using Relational Markov Networks. In: Proc. IJCAI05, Morgan Kaufmann, Inc. (2005) 773–778
9. Pentney, W., Popescu, A., Wang, S., Kautz, H., Philipose, M.: Sensor-based understanding of daily life via large-scale use of common sense. In: Proc. AAAI06
10. Biswas, R., Thrun, S., Fujimura, K.: Recognizing activities with multiple cues. In: IEEE Works. on Human Motion. (2007) 255–270
11. Bobick, A., Ivanov, Y.: Action recognition using probabilistic parsing. In: CVPR98. (1998) 196–202
12. Kok, S., Singla, P., Richardson, M., Domingos, P.: The Alchemy system for statistical relational AI. Tech Report - CS Dept., Univ. of Washington (2005)
13. Minker, J., Seipel, D.: Disjunctive logic programming: A survey and assessment. In: Computational Logic, London, UK, Springer-Verlag (2002) 472–511
14. Singla, P., Domingos, P.: Memory-efficient inference in relational domains. In: Proc. AAAI06, AAAI Press (2006)
15. Singla, P., Domingos, P.: Entity resolution with Markov Logic. In: Proc. ICDM, IEEE (2006) 572–582
16. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. CVPR05. Volume 2., IEEE (2005) 886–893
17. Liu, H.: The ConceptNet Project - http://web.media.mit.edu/∼hugo/conceptnet/
18. Stork, D.: The OpenMind Initiative - http://www.openmind.org/

A Logic Rules

List of rules and their corresponding weights used for the parking lot problem (see problem description in Sec. 3).

1. If a person disappears, he/she enters a nearby car, $w = \frac{4}{5}MAXW$
   $disappear(h) \rightarrow (\exists c\ (inLeftZone(c,h) \vee inRightZone(c,h)) \wedge enter(c,h))$
2. If a person opens the trunk of a car, he/she will (likely) enter that car
   $disappear(h) \wedge openTrunk(c,h) \rightarrow enter(c,h)$, $w = \frac{4}{5}MAXW$
3. A person enters only one car: $enter(c_1,h) \wedge \neg equal(c_1,c_2) \rightarrow \neg enter(c_2,h), w = MAXW$ (if h gets into c temporarily and gets out of it, h is not considered to have entered c, just being temporarily occluded)
4. A person entering a car c from the left (driver) side will (likely) drive c
   $inLeftZone(c,h) \wedge enter(c,h) \rightarrow drive(c,h)$, $w = \frac{4}{5}MAXW$
5. A person entering a car c from the right (passenger) side will (less likely) drive c
   $inRightZone(c,h) \wedge enter(c,h) \rightarrow drive(c,h)$, $w = \frac{1}{5}MAXW$
6. Two persons shaking hand with each other will (likely) not enter the same cars.
   $shakeHand(h_1,h_2) \wedge enter(c_1,h_1) \rightarrow \neg enter(c_1,h_2)$, $w = \frac{4}{5}MAXW$
7. For a car to drive away, it needs a driver: $carLeave(c) \rightarrow (\exists\ h\ \ enter(c,h) \wedge drive(c,h))$, $w = MAXW$
8. A car has only one driver: $drive(c,h_1) \wedge \neg equal(h_1,h_2) \rightarrow \neg drive(c,h_2), w = MAXW$
9. A person can drive only one car: $drive(c_1,h) \wedge \neg equal(c_1,c_2) \rightarrow \neg drive(c_2,h), w = MAXW$
10. A person has to enter a car to drive it: $drive(c,h) \rightarrow enter(c,h), w = MAXW$