Keywords: surveillance system, multi-target tracking, activity recognition, particle filter, probabilistic relational models.

Abstract

The explicit recognition of the relationships between interacting objects can improve the understanding of their dynamics.

In this work, we investigate the use of Relational Dynamic Bayesian Networks to represent the interactions between moving objects in a surveillance system. We use a transition model that incorporates First-Order Logic relations and a two-phases Particle Filter algorithm in order to directly track relations between targets.

We present some results about activity recognition in monitoring coastal borders.

1 Introduction

An activity recognition system aims at recognizing the actions and the goals of one or more agents from a series of observations on the agents’ positions, attributes and the environmental conditions. An activity recognition system consists of multiple modules:

1. The motion detection module processes data from different types of sensor, detecting the objects in the scene that are moving significantly.

2. In the tracking module the objects detected in the previous step are associated with their “path”. If only one object is moving in the scene, this task is straightforward. If there are multiple objects in the scene, the task becomes more complex and a data association step is required. Data association algorithms find the best solution to the problem of associating each moving object in the scene with its sequence of positions. If the objects are interacting, the complexity of the tracking task grows considerably.

3. The activity recognition module associates each (or groups) of the detected paths with a particular activity, giving meaning to the motion of the objects. Activity recognition can be exploited on-line with the tracking (i.e., at each time step, when the measurement is acquired, the state of the domain is filtered and the belief over the activity is corrected) or when a sufficient amount of knowledge about the domain has been acquired. Exploiting the activity recognition task off-line, allows the use of models to match which activity has occurred in the scene.

In some cases, applications are also required to raise an alarm when a particularly dangerous or forbidden situation arises, these systems are called Anomaly Detection Systems. The most part of the models used for activity recognition can be used also with the purpose of anomaly detection in order to decide when to signal an alarm. Anomaly detection is the problem of detecting the occurrence of suspicious events. Activity recognition methods can detect anomalies when these suspicious events are known or described by a model representing interactions among multiple objects moving in a scene.

In this paper we use probabilistic relational models to both improve tracking thanks to prior beliefs about the ongoing activity, and feed-back from the knowledge acquired about the state, to the activity recognition module to compute beliefs over the activity on-line.

2 Motivations

Context interpretation and context-based reasoning have been shown to be key factors for Computer Vision in the development of algorithms for object recognition [4]. In this domain the context is the scene where objects are located and the knowledge about it is expressed by the beliefs over the scene [5]. In this paper we deal with moving objects and we exploit context, i.e., “what is happening around the object we are tracking”. Knowing the scene context can improve the object recognition task, and the knowledge about the identity of the objects improves the belief over the scene; knowing what is happening in the scene (which “relations” are believed to be true in the scene) can improve the tracking and the knowledge about the state of the objects can improve our knowledge about the relations between the objects in the scene (i.e. the context).

Consider, for example, the situation in which we have a group of people walking in a park. If we know they are walking together (i.e. if we have a certain belief over their relation), we know they will exhibit similar behaviors or similar motions. This will help us in tracking them. Moreover, taking into account the relations between objects can also allow
us to recognize complex activities like, for example, the activity of “going to a pub together”: single persons walking can be a simple fragment of a more complex activity that includes some people meeting, going in the same direction, waiting together at different points, and entering into the pub. Dealing with relations between moving objects allows us to recognize a complex activity like this one from another similar one that might be “catching the subway during rush hour”: this complex activity also includes a group of people walking together in the same direction but those people will not wait for each other. In the last years computer vision has mostly dealt with the recognition of single-person, short-duration activities [13]. Here we are interested in more complex activities that involve relations between objects and/or single actions during longer time.

In our work we model the context as a set of First-Order Logic (FOL) relations using them in two principal ways:

- We use relations to improve the efficiency of the tracking. The information contained in the relationships can improve prediction, resulting in better estimation of object trajectories.

- We monitor relations as a goal in itself. This is the case in many applications like traffic prediction or consumer monitoring, anomaly detection or activity recognition.

In this work we consider Relational Dynamic Bayesian Networks (RDBNs), an extension of Probabilistic Relational Model [6] to dynamic domains, as a formalism to monitor relations between moving objects. In our RDBN-based model, relationships are considered as random variables whose values change over time. While tracking the objects in the domain, we also track the evolution of their relationships. For this purpose, in the next sections we formalize a dynamic model able to predict the future state of the objects taking into account their relations, and we describe our Relational Particle Filter that adapts to these settings. We apply this method to the problem of the surveillance of a harbor, analyzing the data obtained from the Intelligent System Challenge 2008-2009 (http://www.intelligent-systems-challenge.ca/home/index.html). After a brief review of the literature, we conclude with some final remarks.

3 Modeling and Inference

A relational domain is a set of objects with relations between them. We will call the state $s$ of a relational domain the relational state. A relational state is the set of the attributes of all the objects and their relations in the domain. Therefore, we can divide the relational state into two parts: the state of the attributes $s^a$ and the state of the relations $s^r$. We will write: $s = [s^a, s^r]$.

A Relational Bayesian Network (RBN) is a directed graph whose nodes are First-Order Logic predicates (both attributes and relations are called predicates in FOL) in the relational domain and whose structure represents the causality between the nodes.

When we deal with dynamics, relational states evolve with time and RBNs have to be extended to RDBNs. A Relational Dynamic Bayesian Network is structured as a pair of RBNs $(B_0, B_{-\infty})$, where $B_0$ represents the probability distribution over the state of the relational domain at time 0 and $B_{-\infty}$ is a RBN of nodes at time $t$ whose parents are predicates at time $t - 1$ or predicates at time $t$ and nodes at time $t - 1$ without their parents.

In order to perform inference in a multi-target setting, we need to extend the algorithms traditionally used in tracking to represent relations. As in classic tracking, the aim is to estimate the current posterior distribution over the state space $s_t$ conditioned on the sequence of observations $z_{1:t}$ up to time $t$: $p(s_t|z_{1:t})$. This distribution is called the target’s belief (bel($s_t$)).

The tracker predicts the probability distribution of the future state $s_{t+1}$, given the knowledge about the current state $s_{t-1}$, by means of a state transition model $p(s_t|s_{t-1})$. Once measurements about the state at time $t$ ($z_t$) are acquired, the state is filtered using the sensor model $p(z_t|s_t)$ that relates (potentially noisy) measurements to the state.

To deal with the problem of inference in a relational domain we have to introduce relations in the following components:

**The transition model** $p(s_t|s_{t-1}) = p(s_t^a, s_t^r|s_{t-1}^a, s_{t-1}^r)$ is a joint probability of the state of all attributes and relations. We assume that the state of relations is not directly affected by the state of the attributes at the previous time step (Fig. 1). Therefore the transition model can be rewritten as:

$$p(s_t^a, s_t^r|s_{t-1}^a, s_{t-1}^r) = p(s_t^a|s_{t-1}^a, s_{t-1}^r)p(s_t^r|s_{t-1}^r). \quad (1)$$

**The sensor model** $p(z_t|s_t) = p(z_t|s_t^a, s_t^r)$ gives the probability of the relational state at time $t$ given the measurements obtained at the same time. We assume the relations are not directly measurable, so the observation $z_t$ is independent of the relations between objects:

$$p(z_t|s_t^a, s_t^r) = p(z_t|s_t^r). \quad (2)$$

![Figure 1. Relational Transition Model. Arrows indicate probabilistic dependence between variables.](image-url)
Assuming the completeness of the state and the conditional independence of the measurements given the relational state, we can use a Bayesian filter algorithm to compute the belief of the relational state. A Bayesian filter algorithm computes the belief of the state at time $t$ ($bel(s_t)$) recursively from the belief of the state at the previous time step. It computes:

$$bel(s_t) = \alpha p(z_t|s^a_t) \tilde{bel}(s_t)$$  \hspace{1cm} (3)

where $\alpha$ is a normalization constant and $\tilde{bel}(s_t)$ is the prediction done over the system $(p(s^a_t, s^r_t|z_{1:t-1}))$ that can be computed as:

$$\tilde{bel}(s_t) = \int p(s^a_t, s^r_t|s^a_{t-1}, s^r_{t-1})bel(s_{t-1})ds_{t-1}.$$  \hspace{1cm} (4)

According to the state transition model (Eq. 1), we can write Eq. 4 as:

$$\tilde{bel}(s_t) = \int p(s^a_t|s^a_{t-1}, s^r_{t-1})p(s^r_t|s^a_{t-1}, s^r_{t-1})bel(s_{t-1})ds_{t-1}.$$  \hspace{1cm} (5)

We will represent the two partial transition models of Eq. 1 by a First Order Decision Diagram (FODD) [2] and introduce an example of it in Section 4.

### 3.1 Relational Particle Filter

The specific and complex probabilistic nature of the model makes it impossible to use filters that require a probabilistic function in closed form, such as the Kalman filter. To solve this issue we developed an extension of the Particle Filter (PF) algorithm.

The PF algorithm [1] is a Monte Carlo method that approximates the required posterior density function by a set of random samples with associated weights, and computes estimates based on these samples and weights. As the number of samples becomes very large, the Monte Carlo approximation to the correct posterior improves and the PF approaches the optimal Bayesian estimate.

Algorithm (1) integrates the relational transition model of Equation (1) in a two-phase PF algorithm called the Relational Particle Filter (RPF).

A particle $(x_{t,(m)})$ is a representation of the state. For this reason, in our setting, it is divided in two parts: the part of the attributes $x^a_{t,(m)}$ and the part of the relations $x^r_{t,(m)}$. The part of the particle relative to the attributes is sampled according to $p(s^a_t|s^a_{t-1}, s^r_{t-1})$ (Line 1). Subsequently, the part of the particle relative to the relations is sampled according to the second part of the relational transition model (Line 2). When the measurement is acquired, particles are weighted according to the sensor model (Line 3). The sensor model takes into account only the part of the particles relative to the attributes, since the particles are composed by two parts, also the parts associated to the relations are weighted. After the weighting step, weights are normalized (Line 4) and the set of particles for the next iteration is extracted according to the normalized weights in the resampling step (Line 5).

The intuition behind this algorithm is the following. At each time step we have samples constituting an approximation of $p(s^a_t|s^a_{t-1}, s^r_{t-1})$ and we want to approximate $p(s^a_t, s^r_t|s^a_{t-1}, s^r_{t-1})$ with a new set of samples. Since the transition model is such that $p(s^a_t, s^r_t|s^a_{t-1}, s^r_{t-1}) = p(s^a_t|s^a_{t-1}, s^r_{t-1})p(s^r_t|s^a_{t-1}, s^r_{t-1})$ we can obtain samples $[s^a_t, s^r_t]$ by augmenting each of the existing samples $[s^a_{t-1}, s^r_{t-1}]$ by sampling $s^a_t$, $s^r_t$ from $p(s^a_t|s^a_{t-1}, s^r_{t-1})$, $p(s^r_t|s^a_{t-1}, s^r_{t-1})$ with the new state of relations $s^r_t$.

### 4 Experiments

We evaluated the proposed method on a synthetic data set provided by the Intelligence System Challenge 2008-2009. We chose to use this data set because it is easier than a real data set, but still challenging because the data were created to accurately reflect the problems faced by those who analyze real data. The data set contains the description of 40 events involving boats at sea. The events described can be either rendezvous, pickUp or avoidance. Given the few number of pickUp events in the data set we decided to consider only the other two events (in this way we are left with 37 events). The data set is composed by 19 rendezvous and 18 avoidances. Rendezvous is the activity of two boats that stop or travel slowly together to exchange goods, all the other events are classified as avoidance. The boats involved in each event can be classified as yacht, shipper or cargo boat with respect to their dimension, speed and frequency of heading change. Each element of the data set reports the tracks of the two boats participating in an event together. At each time interval at most one event takes place.

After describing the setting of our experiments, in the next subsection, we report some results.

#### 4.1 Settings and Results

Observing the rendezvous events in the data set we noticed a peculiar characteristic of the two boats involved in the encounter: one plays the role of master, while the other that of
Figure 2. Example of Rendezvous. Of each boat the x and y coordinate and the coordinate for the speed are reported.

Figure 3. Example of Avoidance.

slaves. In particular, we focused on the variation of speed of the two targets: the master-boat is the one that first decreases its speed and decides where to stop (or start going very slowly) and when the encounter is finished; the slave-boat “imitates” the behavior of the previous ship (Fig. 2). Different is the case of ships that are avoiding each other (thus not in relation according to our model), one maintains its speed and the other decelerates (Fig. 3).

We used the data set to estimate the prior model for the event rendezvous between different couples of boats and the prior for a boat involved in a rendezvous to be the master-boat. From the plots in Fig. 2 and in Fig. 3 it is also possible to notice the four-phases which characterize the event of rendezvous:

1. ships are traveling independently in the phase zero,
2. they approach each other reducing both their speed in the first phase,
3. they travel in the same direction with nearly-zero speed in the second phase and
4. finally they go apart and at least one of them changes its speed in the last phase.

Our (relational) transition model takes into account these different phases allowing us to detect when the event starts and when it finishes.

We represent the state of the attributes with the position \( p \) and the velocity \( v \) of each target in the scene and use a dynamic model that computes \([p_t, v_t]\) as the following:

\[
p_t = p_{t-1} + v_{t-1} dt + \frac{1}{2} a dt^2 \tag{6}
\]

\[
v_t = v_{t-1} + a dt, \tag{7}
\]

where \( a \) is a random variable whose distribution depends by the object type, the relation and, if the relation is true, it depends also by the role played by the object and the phase of the ongoing activity and whose distribution can be given by the FODD reported in Fig. 4.

Each particle represents the state of the domain: it represents the positions and the velocities of all the objects in the scene as well as their relations. When sampling the particles, we reason about each object to predict their future state taking into account the relations between itself and the other objects. An examples of the FODD used to model the transition of the state of the relations is reported in Fig. 5.

We ran the experiments on each of the 37 events in the data set. We apply PF techniques with \( 10^5 \) particles. In the first two rows of Table 1 we show the accuracy of our method for rendezvous detection compared to the accuracy of a method that randomly chooses which boats are in that relation. We compute the tracking error as follow: we first compute for each object the distance between the real trajectories and the trajectories estimated by the RPF averaged for all the particles, we, then, average this errors over all the time steps. In this way we obtain the evarage error of tracking of an object.

We ran the experiments on each of the 37 events in the data set. We apply PF techniques with \( 10^5 \) particles. In the first two rows of Table 1 we show the accuracy of our method for rendezvous detection compared to the accuracy of a method that randomly chooses which boats are in that relation. We compute the tracking error as follow: we first compute for each object the distance between the real trajectories and the trajectories estimated by the RPF averaged for all the particles, we, then, average this errors over all the time steps. In this way we obtain the evarage error of tracking of an object.
results obtained with our method (RPF) and a standard method (PF) that does not take into account correlations between objects) in the last rows of Table 1 where the error is reported for RPF and PF for all the tracks and for those tracks that RPF correctly recognizes as rendezvous activity.

From the table we can see that (1) the accuracy of our relational activity recognition method is better than the simple method (2) on average our method approximates that tracks correctly recognizes as rendezvous.

5 Related works

Our work is at the intersection of works in Probabilistic Relational Models, that to our knowledge have never before considered applications in tracking and computer vision, where often heuristics are used to improve tracking, but not with a systematic account of relationships between targets.

Recently there has been increasing interest in models that extend probabilistic reasoning to FOL to exploit redundancies observed in the worlds [6], [7]. In this setting, many relational inference algorithms proceed by first fully instantiating the First-Order relations and then working at the propositional level. In [11] an inference algorithm that instantiates relations only as needed is presented, but this algorithm can deal only with static domains as the relations are not supposed to change over time. Moreover, our model is different from the one presented in [10], where the concept of class is used to develop an inference system able to deal with a large number of heterogeneous objects. We use FOL to explicitly represent relationships between objects to improve the inference task. Our method is potentially applicable to situations with a large number of objects as well.

Mixed-states models have been used to deal with complex tracking tasks [8]. They use a mixed-state object representation combining continuous-valued shape parameters with a discrete label that encodes which of a discrete set of motion models is in force. While the inference task is performed, the discrete label says which model must be used to predict the future (continuous) state. In the prediction step, model switching is performed when necessary.

In a mixed-states models the label of the transition model relates the prediction of the future state only to the time step previous to the current one, this assumption makes the prediction over the motion model unrelated with the current state, this may lead to a possible delay for the tracker to correctly approximate the objects’ behavior. In our method, instead of discrete labels, we use relations as the representation of the motion model in force. The state of relations does depend on the previous relational state and on the current state of attributes: the prediction step takes into account which relations are true to make its hypothesis over the next state, that results in being more up-to-dated with respect to a mixed-state model approach. Moreover, the use of FOL relations (as opposed to a list of possible motion models) generalizes our models to different domains.

In [9] the recognition of complex activity (temporally extended activities that can be fragmented in simple ones) is based on context-free grammar. They decouple the recognition task in two levels: a lower level that detects simple activities that define the inputs for the stochastic context-free grammar used as a “bag of words” for a parsing mechanism. Instead, our approach does not decouple the recognition task. It seeks to take advantage from the tracking, that provides the detection of simple activities, to recognize the temporally extended activity and from the knowledge about the complex activity to improve the tracking.

In [12], the authors address the problem of activity recognition using FOL rules and Markov Logic Networks to represent common sense domain knowledge. Differently from the method we propose, the inference task is performed off-line: they perform probabilistic inference for input queries about events of interest already happened. We seek, instead, to perform an on-line probabilistic inference of both the state of the domain and the activities.

We are aware of the fact that in the proposed approach relationships must be handcrafted but we think that the proposed method will perform well in real-word scenarios as well, where there are more than two objects interacting with each other and the interactions can be more complicated. Consider the domain presented in [3], where multiple bycicles can be drop off

<table>
<thead>
<tr>
<th></th>
<th>RPF</th>
<th>PF</th>
<th>random</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>0.7368</td>
<td>0.4444</td>
<td></td>
</tr>
<tr>
<td>TN</td>
<td>0.6667</td>
<td>0.4841</td>
<td></td>
</tr>
<tr>
<td>Error (m, 37 tracks)</td>
<td>1.7359</td>
<td>1.8460</td>
<td></td>
</tr>
<tr>
<td>Error (m, 14 tracks)</td>
<td>1.4326</td>
<td>2.5431</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5. An example of FODD for $p(s_t|s_{t-1}, s_t^r)$. At each time step, for each object it computes the probability of the object to be in relation (or not) with another object given their attributes and the distance between them.
or picked up by persons. With the right assiomatic structure (as the one presented in the cited paper) our method can model those complex interactions and reasoning about their evolution during time.

6 Conclusions

In this paper we presented a technique based on relational Bayesian reasoning in order to address the problem of activity recognition and tracking. We used an extension of particle filter, called relational particle filter, to make inference in relational domains. We have shown how using relations as context can improve the tracking task for surveillance purposes. From our results we can conclude that our method can help to identify the type of encounter that the targets are engaging.

Moreover, the master-slave relation we dealt with can be very important for other different domains: it can be useful when we want to detect if a person is following another person or to detect who is the “leader” of a group of people and for many other situations.

Acknowledgements

The authors are aware of the works of Sanghai, Weld and Domingos on RDBNs; however the paper presenting their work has been retracted. Refer to: http://www.aaai.org/Library/JAIR/Vol24/jair24-019.php

References


