

Relations as Context to Improve Multi-Target Tracking and Activity Recognition

Cristina Manfredotti^{1,2}, Enza Messina¹, David Fleet²

¹ DISCo, University of Milano-Bicocca,
{manfredotti, messina}@disco.unimib.it
² Computer Science Dept, University of Toronto,
{cristina, fleet}@cs.toronto.edu

Abstract. The explicit recognition of the relationships between interacting objects can improve the understanding of their dynamic model.

In this work, we investigate the use of Relational Dynamic Bayesian Networks to represent the dependencies between objects behavior in the context of multi-target tracking. We propose a new formulation of the transition model that accommodates for First-Order Logic relations and we extend the Particle Filter algorithm in order to directly track relations between targets.

Many applications can benefit from this work, as activities recognition, traffic monitoring, strategic analysis, sports coaching and others. We present some results about activity recognition in monitoring Canadian costal borders.

Key words: Multi Target tracking, Probabilistic Relational Models, Bayesian Filtering, Particle Filtering.

1 Introduction and 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 [3]. 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 [4].

In this paper we deal with moving objects and we refer the concept of context to “what is happening around the object we are tracking”. Knowing the scene can improve the objects 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 relation 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 have a similar behavior or a similar motion. 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

each other at different points and entering together into the pub. Dealing with relations between moving objects allows us to recognize a complex activity like this one from another similar one that can be the “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 activities composed by the repetition of simple movements [12], instead those are examples of more complex activities that involve relations between objects and/or single actions during time.

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

- We will use *relations to improve the efficiency of the tracking*. The information contained in the relationships can improve prediction, resulting in a better estimation of objects trajectories.
- We will *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)¹, an extension of *Probabilistic Relational Model* [5] 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 may 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 propose a formalization of a new dynamic model able to predict the future state of the objects taking into account their relations and we introduce a new version of Particle Filter (that we call *Relational Particle Filter*) that adapts to these settings. After presenting some preliminary results obtained on the Intelligent System Challenge 2008-2009 data set², and a brief review of the literature, we conclude with some final remarks.

2 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 *relational state*, and we define it as the set of instantiations of all the objects and their relations in the domain. Therefore, we can divide the relational state in two parts: the *state of the objects*³ s^o and the *state of the relations* s^r and we will write: $s = [s^o, s^r]$.

A *Relational Bayesian Network* (RBN) is a directed acyclic graph whose nodes are First-Order Logic attributes or relations between objects in the relational domain and whose structure represents the causality between the nodes.

¹ 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/Vol124/jair24-019.php>

² <http://www.intelligent-systems-challenge.ca/home/index.html>

³ We will use the terms *state of objects* and *state of instantiations* interchangeably.

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

In order to make 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 of the state space s_t conditioned to the sequence of observations $z_{1:t}$ up to time t : $p(s_t|z_{1:t})$. This distribution is often called the *target's belief* ($bel(s_t)$).

The tracker *predicts* the probability distribution of the future state s_t , 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.

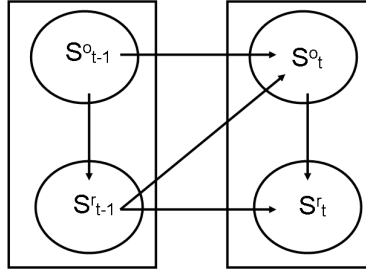


Fig. 1. Relational Transition Model. Arrows indicate probabilistic dependence between variables.

To extend the traditional tracking algorithms to represent relations we introduce the following components:

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

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

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

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

Under the Markov assumption and the conditional independence of the data given the state, we can use a *Bayesian filter algorithm* to compute the belief of the relational

state:

$$bel(s_t) = \alpha p(z_t | s_t^o) \widetilde{bel}(s_t) \quad (3)$$

where α is a normalization constant and $\widetilde{bel}(s_t)$ is the prediction done over the system ($p(s_t^o, s_t^r | z_{1:t-1})$) that can be computed as:

$$\widetilde{bel}(s_t) = \int p(s_t^o, s_t^r | s_{t-1}^o, s_{t-1}^r) bel(s_{t-1}) ds_{t-1}. \quad (4)$$

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

$$\widetilde{bel}(s_t) = \int p(s_t^o | s_{t-1}^o, s_{t-1}^r) p(s_t^r | s_{t-1}^r, s_t^o) bel(s_{t-1}) ds_{t-1}. \quad (5)$$

In the most general case we can represent the two partial transition models of Equation (1) by a First Order Logic Tree (FOPT)⁴. We will introduce an example of FOPT when dealing with the experiments.

2.1 Relational Particle Filter

The specific and complex probabilistic nature of the presented setting makes 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 whose properties are appealing for our case.

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.

We integrate the relational transitional model introduced in Equation (1) in a new *Relational Particle Filter* (RPF), shown in Algorithm (1).

Algorithm 1: Pseudo Code for the Relational Particle Filter algorithm

$bel(s_t) = RPF(bel(s_{t-1}), z_t)$

for all $m = 1 : M$ **do**

1. $x_{t,(m)}^o \sim p(s_t^o | s_{t-1}^o, s_{t-1}^r)$; hypothesis for the state of instantiations
2. $x_{t,(m)}^r \sim p(s_t^r | s_{t-1}^r, s_t^o = x_{t,(m)}^o)$; hypothesis for the state of relations
3. $\omega_{(m)} = p(z_t | x_{t,(m)}^o)$; weights computation

for all $m = 1 : M$ **do**

4. $\widetilde{\omega}_{(m)} = \frac{\omega_{(m)}}{\sum_{m=1}^M \omega_{(m)}}$; weights normalization
 5. Resample $bel(s_t)$ from $\{[x_{t,(m)}^o, x_{t,(m)}^r]\}$ according to weights $\{\widetilde{\omega}_{(m)}\}$ with repetition.
-

⁴ A FOPT (also known as First Order Decision Diagram [2]) is Probabilistic Tree whose nodes are First Order Logic formulas.

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 objects $x_{t,(m)}^o$ and the part of the relations $x_{t,(m)}^r$. (see Figure 2(a)). The part of the particle relative to the instantiations is sampled according to $p(s_t^o | s_{t-1}^o, s_{t-1}^r)$ (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 objects, 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).

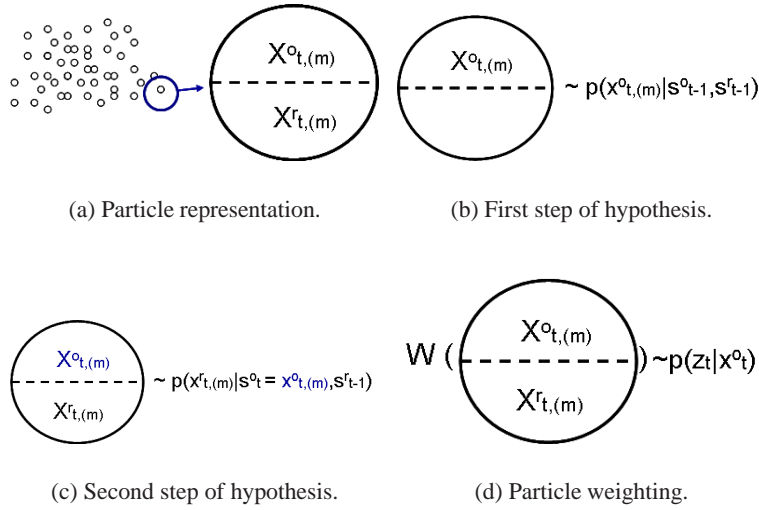


Fig. 2. Cartoon representation of the proposed algorithm.

3 Experiments

We evaluated the proposed method on a synthetic data set that is based on the Intelligence System Challenge 2008-2009 data set. We chose to use this data set because it is easier than a real data set but still challenging. The data set contains the description of the events happened in the sea; each element of the data set reports the tracks of two boats participating in an event (i.e. Rendezvous, PickUp and Avoidance) together. At each time interval at most one event takes place.

We are particularly interested in the case where there is uncertainty about the participants taking part in an event, in order to demonstrate the advantage of maintaining

beliefs over the set of possible relations. In order to test our relational particle filter algorithm for activity recognition in a more challenging scenario with multiple targets, we use the original data set to build a new synthetic data set of 120 situations (either rendezvous or avoidance), obtained by pairing two encounters randomly sampled from the original data set. In this way, four ships are present at the same time in the scene.

In the experiment, we consider the task of detecting a rendezvous between a yacht and a fisher ship. After describing the setting of our experiments, in the next subsection, we report some results.

3.1 Settings and Results

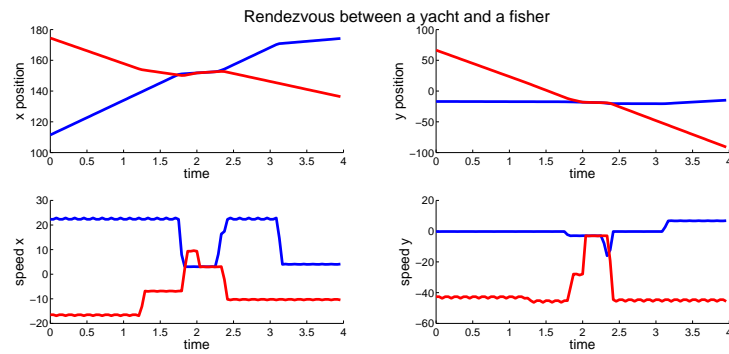


Fig. 3. Example of Rendezvous.

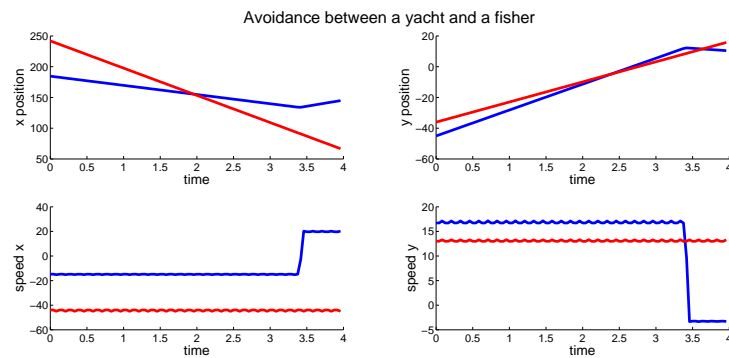


Fig. 4. Example of no relation.

We used the data set to estimate the prior for the event Rendezvous between a Fisher and a Yacht (33/80). Then, we examined the data relative to the encounters in order to acquire information about the two different events (rendezvous or avoidance) that can be used to predict the relation. In particular, we focused on the variation of speed of the two targets. Consider for example, the rendezvous in Figure (3): the two ships come closer and both progressively reduce their speed until a nearly-zero value. 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 (Figure (4)).

From the images it is also possible to notice the three-phases which characterize the event of rendezvous: ships approach each other reducing their speed in the first phase, they travel in the same direction with nearly-zero speed in the second phase and finally they go apart and at least one of them change its speed. Our relational transition model takes into account these three different phases allowing to detect when the event starts and when it finishes but also allowing to understand if two ships can be in relation (since a ship can be in relation only with another ship).

An example of the relational transition model used in our experiments is given in Figure (5) and in Figure (6) .

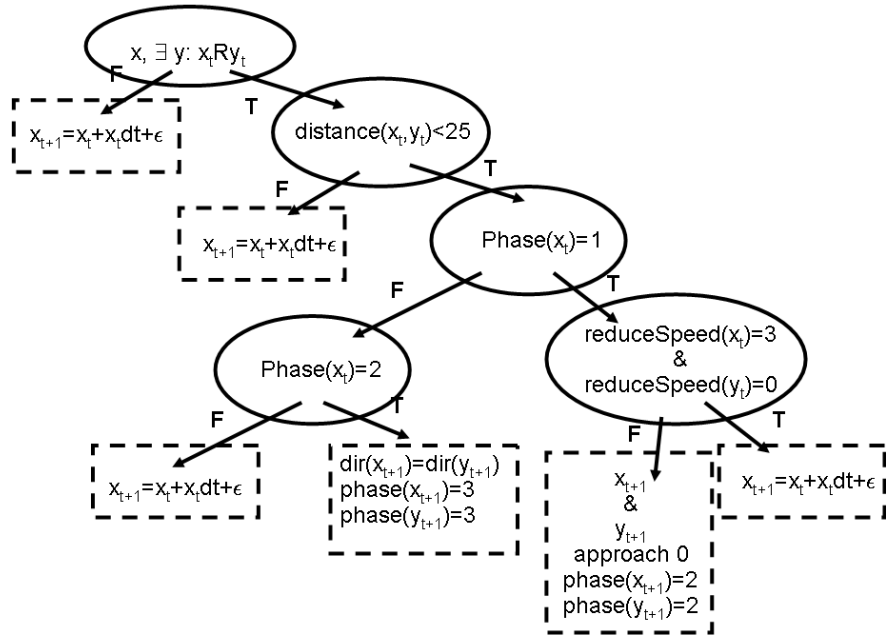


Fig. 5. FOPT for $p(s_t^o | s_{t-1}^o, s_{t-1}^r)$.

We ran the experiments on each of the 120 sets of four tracks in the data set. In table (1) we show the accuracy of our method for the rendezvous detection compared

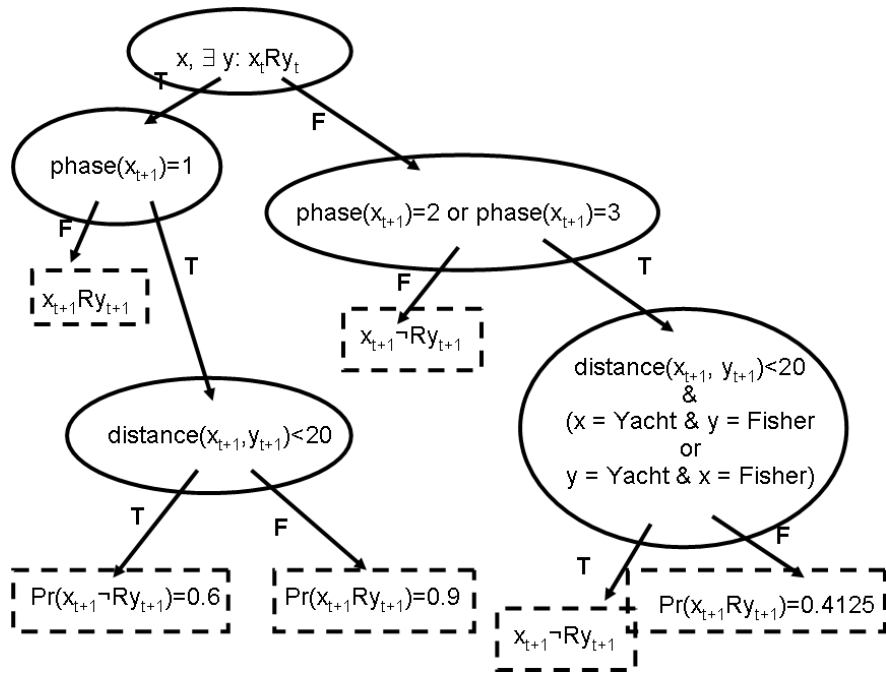


Fig. 6. FOPT for $p(s_t^r | s_{t-1}^r, s_t^o)$.

to the accuracy of a method that randomly chooses which boats are in relation. In the table it is also reported the average tracking error of the RPF algorithm compared to a PF algorithm that does not take into account relations. The tracking error is computed as the distance between the trajectories acquired by the particle filter (at each time-step, it averages the position considering the states of all particles) and the real trajectories.

method	TP ratio	TN ratio	Tracking Error(km, mean)
RPF	0.4545	0.7235	1.8379
PF			3.3906
random choice	0.4444	0.4841	

Table 1. In columns TP ratio and TN ratio the ratio over the 120 set of tracks of true positive and true negative is reported for our method and a random choice method. In the last column the average tracking error for our method (RPF) and a method that does not take into account relations (PF) is reported.

3.2 Related Works

Our work is at the intersection of work 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 First Order Logic to exploit redundancies observed in the worlds ([5], [6]). In this setting, many relational inference algorithms proceed by first fully instantiating the First-Order relations and then working at the propositional level. In [10] 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 [9], where the concept of class is used to develop an inference system able to deal with a large number of heterogenous objects. We use First-Order Logic 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.

Hybrid states models have been used to deal with complex tracking tasks [7]. They combined continuous-valued dynamic with a discrete state of the world (context) encoding which switching dynamic is performed jointly with tracking. Our system uses relations as representations of the context of each object instead of the context of the entire world. The explicit recognition of the relations of each object allows us to deal with much more complex tracking tasks. Moreover, the use of First Order Logic (as opposed to predicative logic) generalizes our models to different domains.

In [8] 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 single simple activities that are 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, but seek 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 [11], the authors address the problem of activity recognition using First Order Logic rules and Markov Logic Networks to represent common sense domain knowledge. Differently from the method we are proposing, 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.

4 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 presented an extension of particle filter, called relational particle filter, that can be used to make inference. From our preliminary results we can conclude that our method can help to identify the type of

encounter that the targets are engaging. Moreover we have shown how using relations as context can improve the tracking task. Compared to hybrid state model techniques, we are able to model the problem with a single dynamic model and the state representation is much more compact.

There are a number of possible applications of this approach in problems where there is the need of monitoring a situation from sensed data (video surveillance, homeland-security, etc.) that we are interested to consider for future works.

References

1. S. Arulampalam, S. Maskell, and N. Gordon. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002.
2. S. J. C. Wang and R. Khardon. First order decision diagrams for relational mdps. *JAIR*, 31:431–472, 2008.
3. A. A. E. Derek Hoiem and M. Hebert. Putting objects in perspective. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2137 – 2144, June 2006.
4. G. Elidan, G. Heitz, and D. Koller. Learning object shape: From drawings to images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.
5. N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *IJCAI*, pages 1300–1309, 1999.
6. L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of link structure. *Journal of Machine Learning Research*, 3:679–707, 2002.
7. M. Isard and A. Blake. A mixed-state condensation tracker with automatic model-switching. In *ICCV*, pages 107–112, 1998.
8. Y. A. Ivanov and A. F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):852–872, 2000.
9. B. Milch and S. J. Russell. General-purpose mcmc inference over relational structures. In *UAI*, 2006.
10. H. Poon, P. Domingos, and M. Sumner. A general method for reducing the complexity of relational inference and its application to mcmc. In *AAAI*, pages 1075–1080, 2008.
11. S. D. Tran and L. S. Davis. Event modeling and recognition using markov logic networks. In *ECCV (2)*, pages 610–623, 2008.
12. R. S. Yan Ke and M. Hebert. Event detection in crowded videos. In *IEEE International Conference on Computer Vision*, October 2007.