

# Simultaneous Tracking and Activity Recognition with Relational Dynamic Bayesian Networks

Cristina Elena Manfredotti, David James Fleet,  
Howard John Hamilton, Sandra Zilles

Technical Report CS 2011-1  
March 30, 2011

Copyright © 2011, C.E. Manfredotti, D.J. Fleet, H.J. Hamilton, S. Zilles  
Department of Computer Science  
University of Regina  
Regina, SK, Canada S4S 0A2  
ISBN 978-0-7731-0694-9

---

# Simultaneous Tracking and Activity Recognition with Relational Dynamic Bayesian Networks

---

**Cristina Elena Manfredotti**  
Dept. of Computer Science  
University of Copenhagen  
Copenhagen, Denmark

**David James Fleet**  
Dept. of Computer Science  
University of Toronto  
Toronto, ON, Canada

**Howard John Hamilton**  
Dept. of Computer Science  
University of Regina  
Regina, SK, Canada

**Sandra Zilles**  
Dept. of Computer Science  
University of Regina  
Regina, SK, Canada

## Abstract

Taking into account relationships between interacting objects can improve the understanding of the dynamic model governing their behaviors. Moreover, maintaining a belief about the ongoing activity while tracking allows online activity recognition and improves the tracking task. We investigate the use of Relational Dynamic Bayesian Networks to represent the relationships for the tasks of multi-target tracking and explicitly consider a discrete variable in the state to represent the activity for online activity recognition. We propose a new transition model that accommodates relations and activities and we extend the Particle Filter algorithm to directly track relations between targets while recognizing ongoing activities.

## 1 INTRODUCTION

Multi-body tracking and activity recognition are intimately coupled in video analysis. While tracking is a natural precursor to the detection and analysis of activities, prior knowledge of activities provides useful dynamical knowledge for improving state predictions and identity maintenance. In essence, activities and relations between objects being tracked provide *context* for tracking, in much the same way that scene context has the potential to resolve ambiguities in object recognition (e.g., see (Hoiem, Efron, & Hebert, 2006; Elidan, Heitz, & Koller, 2006)).

We use the term context to refer to *interactions* between objects being tracked and the underlying *activity*, both of which have a major impact on one's ability to predict state evolution in a dynamical model. In particular, we are interested in activities involving multiple objects. Consider, for example, a complex activity like "going to a pub together", comprised of people meeting, going in the same direction, waiting for each other at different points and entering the

pub together. Dealing with relations between moving objects allows us to distinguish a complex activity like this one from others that might be somewhat similar, such as "catching the subway during rush hour". The latter activity also includes several people walking together in the same direction, but they will likely not wait for each other.

Our work extends the concept of relational databases with Bayesian uncertainty (e.g., (Friedman, Getoor, Koller, & Pfeffer, 1999)) to model probabilistically the dynamics of *relations* between objects. Following (Manfredotti & Messina, 2009), we use relations between the objects in two ways:

- *To improve the tracking efficiency.* Information in the relationships can improve dynamical predictions, resulting in a better estimation of object trajectories.
- *To improve activity recognition.* The explicit recognition of relations between objects allows us to recognize activities of interacting objects.

To represent relations between moving objects, we use *Relational Dynamic Bayesian Networks* (RDBNs) (Manfredotti, 2009)<sup>1</sup>, an extension of *Probabilistic Relational Model* (Friedman et al., 1999) to dynamic domains. In RDBN, relationships are considered as random variables whose values may change over time. While tracking the objects in the domain, we track the evolution of their relationships. In this way, we simultaneously do interacting objects' activity recognition and multi-target tracking.

The main contributions of this paper are:

- A new definition of *state of the world* as the set of the attribute values and the relations of the objects in the world and the formalization of a new dynamic model that takes into account the relations between the objects in the world;

---

<sup>1</sup>The authors are aware of the work by S. Sanghai, P. Domingos, and D. Weld, where RDBNs were first introduced. Sanghai et. al have requested their work to be retracted, as explained at this url: [www.aaai.org/Library/JAIR/Vol24/jair24-019.php](http://www.aaai.org/Library/JAIR/Vol24/jair24-019.php)

- The introduction of a new inference algorithm (called *Relational Particle Filter*) able to predict the future state of the world taking into account the relations and the activities of moving objects;
- Empirical evidence of the usefulness of RDBNs for multi-target tracking and online activity recognition.

## 2 BACKGROUND

The problem we aim at is twofold. On the one hand, there is the problem of simultaneously tracking the objects in a scene and recognizing what they are doing. To address this problem, hybrid state models (Isard & Blake, 1998) combine continuous-valued dynamic with a discrete state of the world. Their discrete state represents the activity encoding which, switching dynamic can be performed jointly with tracking. On the other hand, we want to recognize *complex* activities, which result from the interaction between objects in the scene. The activities that can be recognized by hybrid-state models are mainly activities of single objects (e.g., a bouncing ball) or simple activities that do not take into account interactions between objects.

With our focus on inference problems and relations, our work is at the intersection of research on Probabilistic Relational Models, which to our knowledge have never before been applied to dynamic domains, and Computer Vision, where heuristics are often used to improve tracking or activity recognition, 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 world (Friedman et al., 1999; Getoor, Friedman, Koller, & Taskar, 2002). In these settings, many relational inference algorithms proceed by first fully instantiating the first order relations and then working at the propositional level. Poon, Domingos and Sumner (2008) present an inference algorithm that instantiates relations only as needed. All these algorithms can deal only with static domains because the relations are not supposed to change over time. Milch and Russell (2006) use the concept of class to develop an inference system that deals with a large number of heterogeneous objects. They do not consider relations as interactions between objects that can change the activity the objects are jointly involved in. We use the concept of object class to explicitly represent relationships between objects to improve the inference task.

Ivanov and Bobick (2000) present an approach where the recognition of temporally extended activities is based on context-free grammars. They decouple the recognition task in two subtasks. They first detect single, simple activities that can be treated as inputs for the stochastic context-free grammar used to recognize, as second task, more complex activities. We do not decouple the recognition task, but

seek to make use of the tracking to aid in recognizing complex activities and make use of the knowledge about the complex on-going activity to improve the tracking. Moreover, the complex activities we are interested in are not (only) sequences of simple activities but interactions between objects, and we want to recognize these interactions.

Tran and Davis (2008) formulate the activity recognition task using first order logic rules and Markov Logic Networks to represent common sense domain knowledge. In their formulation, the inference task is performed offline: they perform probabilistic inference for input queries concerning events that have already happened. We seek, instead, to perform an online probabilistic inference about both the state of the relational domain and the activities. Consequently, to our best knowledge, no existing method addresses the problem we focus on.

## 3 ACTIVITIES AND BAYES FILTERING

When tracking a single object, one defines the *state of the world* (or *state*)  $s_t$ , as the attributes of interest at time  $t$ , such as the object’s position and velocity. When formulated in terms of Bayesian filtering, *tracking* entails estimating the *belief* of the state, i.e., the probability of the state conditioned on the observation (measurement) history:

$$bel(s_t) \equiv p(s_t | z_{1:t}), \quad (1)$$

where  $z_{1:t}$  is the sequence of observations up to time  $t$ . The conventional Bayesian filtering equations can be derived with the usual Markov model and the conditional independence of observations on the state.

One can augment the state with activities, yielding a hybrid state, i.e.,  $[a_t, s_t]$ , where  $s_t$  is the state as above, and  $a_t$  is the *discrete* activity (Black & Fleet, 2000; Isard & Blake, 1998). For example, Isard and Blake (1998) use a hybrid model in which the discrete activity represents the contact state of a bouncing ball, and hence the dynamical model that should be used to predict the future position of the ball.

To formulate Bayesian filtering with a hybrid state model, let  $S_t \equiv [a_t, s_t]$  denote an activity-augmented state with

$$bel(S_t) = p(S_t | z_{1:t}). \quad (2)$$

Assuming  $S_t$  to be *complete* and the conditional independence of the observations given the state, a Bayesian filter computes the belief as:

$$bel(S_t) \propto p(z_t | S_t) \int p(S_t | S_{t-1}) bel(S_{t-1}) dS_{t-1}. \quad (3)$$

Isard and Blake (1998) factor the dynamical model as:

$$p(S_t | S_{t-1}) = p(a_t | a_{t-1}, s_{t-1}) p(s_t | a_t, s_{t-1}), \quad (4)$$

where  $p(a_t | S_{t-1})$  describes the expected change in activity given the previous state and  $p(s_t | a_t, s_{t-1})$  describes the

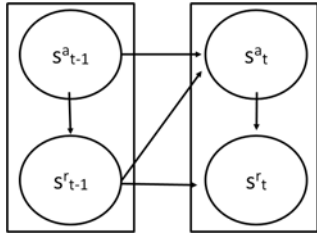


Figure 1: Relational Transition Model. Arrows indicate probabilistic dependence between variables.

conditional evolution of the object’s state given the activity (e.g., the motion of the ball given its current contact state).

## 4 OBJECTS AND RELATIONS

In this paper we focus on multi-object tracking, for which *relations* between the objects being tracked play a central role. Consider, for example, the difference between the activities of “passing the ball” and “intercepting the ball” in the soccer domain; both activities result in a new player having control of the ball, but “passing” requires the two players to be on the same team (i.e., to be in the relation of having the same value for the team attribute) and “intercepting” requires the players to be on different teams.

To take into account relations between objects, we introduce the idea of a *relational domain*. A relational domain is the set of objects in the world and their relations. We call the state  $s_t$  of a relational domain at time step  $t$  *relational state*. A relational state is the set of the attributes of all the objects and their relations in the domain at some time step. We can divide the relational state in two parts, the *state of the attributes*  $s_t^a$  and the *state of the relations*  $s_t^r$ . Accordingly, in what follows we consider states of the form  $s_t \equiv [s_t^a, s_t^r]$  and activity-augmented states  $S_t \equiv [a_t, s_t]$ .

### 4.1 Relational Dynamic Bayesian Networks

Uncertainty in a relational domain can be modeled with *Relational Bayesian Networks* (RBNs). An RBN is a directed graph whose nodes are attributes of objects or relations between objects in a relational domain and whose arcs represents the causality between the nodes (Jaeger, 1997).

To deal with dynamic domains, where relational states evolve with time, RBNs can be extended to *Relational Dynamic Bayesian Networks* (RDBNs). An RDBN is structured as a pair of RBNs ( $B_0, B_{\rightarrow}$ ), where  $B_0$  represents the probability distribution of the state of the relational domain at time step  $t = 0$  and  $B_{\rightarrow}$  models the transition probability of the state.  $B_{\rightarrow}$  is a graph whose nodes represent variables at time  $t$  and their parents. In  $B_{\rightarrow}$  nodes representing variables at time  $t$  can have parents only at time  $t - 1$  or  $t$ .

Accordingly, a probabilistic model for a relational domain requires the definition of a prior over the relational state,  $p(s_0)$ , a dynamical model  $p(s_t|s_{t-1})$ , and a sensor model  $p(z_t|s_t)$ . The *dynamical transition model* describes the probabilistic evolution of all attributes and relations. Without loss of generality, it can be factored as:

$$p(s_t | s_{t-1}) = p(s_t^a | s_{t-1})p(s_t^r | s_t^a, s_{t-1}). \quad (5)$$

As depicted in Fig. 1, we further assume that the relational state is not directly affected by the state of the attributes at the previous time step. Accordingly, Eq. 5 can be simplified as follows:

$$p(s_t | s_{t-1}) = p(s_t^a | s_{t-1})p(s_t^r | s_t^a, s_{t-1}^r). \quad (6)$$

Similarly, for the *activity-augmented relational state*,  $S_t = [s_t, a_t]$ , the dynamical transition model becomes

$$\begin{aligned} p(S_t | S_{t-1}) &= p(a_t | S_{t-1})p(s_t | a_t, s_{t-1}) \\ &= p(a_t | S_{t-1})p(s_t^a | a_t, s_{t-1})p(s_t^r | a_t, s_t^a, s_{t-1}^r) \end{aligned} \quad (7)$$

The *sensor model*  $p(z_t | S_t) \equiv p(z_t | a_t, s_t^a, s_t^r)$ , gives the probability of the observations  $z_t$  at time step  $t$ , given the augmented relational state at the same time step. We assume the relations and the activities are not directly observable. This assumption is reasonable: for example, we can measure the position of a person but we will have to infer from the sequence of his positions that his activity is that of going to a pub. Under this assumption, the observation  $z_t$  is independent of the relations and the activity:

$$p(z_t | S_t) = p(z_t | s_t^a). \quad (8)$$

### 4.2 Relational Particle Filter

In order to perform inference in a relational multi-target settings, we need to extend the algorithms traditionally used to represent relations. Under the Markov assumption and the conditional independence of the observations given the relational state, we can use a Bayesian filter algorithm to compute the belief of the augmented relational state:

$$bel(S_t) \propto p(z_t | s_t^a) \tilde{bel}(S_t), \quad (9)$$

where the *prediction distribution*,  $\tilde{bel}(S_t)$ , is given by

$$\tilde{bel}(S_t) \equiv p(S_t | z_{1:t-1}) = \int p(S_t | S_{t-1}) \tilde{bel}(S_{t-1}) dS_{t-1}$$

Using the above dynamical transition model (Eq. 7), we can rewrite Eq. 9 as:

$$\begin{aligned} \tilde{bel}(S_t) &= \int p(a_t | S_{t-1}) \\ & p(s_t^a | a_t, s_{t-1}) p(s_t^r | a_t, s_t^a, s_{t-1}^r) \tilde{bel}(S_{t-1}) dS_{t-1} \end{aligned} .$$

Given the hybrid nature of the probabilistic model, and the nonlinear nature of most dynamical and sensor models of interest, one cannot expect to find closed form solutions to the filtering equations, like the well-known Kalman update equations. We therefore consider an extension of the Particle Filter (PF) algorithm for the relational model. The PF algorithm (Arulampalam, Maskell, & Gordon, 2002) is a sequential Monte Carlo method that approximates the target posterior distribution with a set of random samples with associated weights and computes estimates based on these samples and weights. As the number of samples becomes 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 Eq. 7 in a three-phase PF algorithm called *Relational Particle Filter* (RPF).

---

**Algorithm 1:** Relational Particle Filter algorithm

---

- $$\{S_t^{[m]}\}_{m=1}^M = RPF(\{S_{t-1}^{[m]}\}_{m=1}^M, z_t)$$
- for all**  $m = 1 : M$  **do**
1. hypothesis for the activity:  
 $a_t^{[m]} \sim p(a_t | S_{t-1} = S_{t-1}^{[m]});$
  2. hypothesis for the state of the attributes:  
 $s_t^{a,[m]} \sim p(s_t^a | a_t = a_t^{[m]}, s_{t-1} = s_{t-1}^{[m]});$
  3. hypothesis for the state of the relations:  
 $s_t^{r,[m]} \sim p(s_t^r | a_t = a_t^{[m]}, s_t^a = s_t^{a,[m]}, s_{t-1}^r = s_{t-1}^{r,[m]});$
  4. compute weights:  $\omega^{[m]} = p(z_t | S_t^{a,[m]});$
- for all**  $m = 1 : M$  **do**
5. normalize weights:  $\tilde{\omega}^{[m]} = \frac{\omega^{[m]}}{\sum_{m=1}^M \omega^{[m]}};$
- for all**  $m = 1 : M$  **do**
6. draw  $i$  with probability  $\propto \tilde{\omega}^{[m]}$  and add  $S_t^{[i]}$  to the set  $\{S_t^{[m]}\}.$
- 

At each time step, we have  $M$  samples  $\{S_{t-1}^{[m]}\}_{m=1}^M$ , which approximate  $bel(S_{t-1})$ , and we want to approximate  $bel(S_t)$  with a new set of samples  $\{S_t^{[m]}\}_{m=1}^M$ . A particle ( $S_t^{[m]} = [a_t^{[m]}, s_t^{a,[m]}, s_t^{r,[m]}]$ ) is a particular hypothesis about the augmented relational state. In our settings, a particle is composed of three parts: the activity  $a_t^{[m]}$ , the attributes  $s_t^{a,[m]}$  and the relations  $s_t^{r,[m]}$ .

The algorithm first builds a temporary particle set which approximates  $\int p(S_t | S_{t-1}) bel(S_{t-1}) dS_{t-1}$ . It processes each single particle  $S_{t-1}^{[m]}$  in the input set and generates a particle  $S_t^{[m]}$  as a sample of the proposal distribution  $\int p(S_t | S_{t-1}) bel(S_{t-1}) dS_{t-1}$ . This particle is obtained by applying the transition model  $p(S_t | S_{t-1})$  to the particle  $S_{t-1}^{[m]}$  which represents a sample of  $bel(S_{t-1})$ . Since the transition model is such that  $p(S_t | S_{t-1}) = p(a_t | S_{t-1}) p(s_t^a | a_t, s_{t-1}) p(s_t^r | a_t, s_t^a, s_{t-1}^r)$ , we can obtain samples  $[a_t^{[m]}, s_t^{a,[m]}, s_t^{r,[m]}] \sim p(a_t, s_t^a, s_t^r | a_{t-1}, s_{t-1}^a, s_{t-1}^r) bel(S_{t-1})$  applying first

the model  $p(a_t | a_{t-1}, s_{t-1}^a, s_{t-1}^r)$  to the particle  $S_{t-1}^{[m]}$  (Line 1) and then augmenting the obtained sample  $[a_t^{[m]}, s_{t-1}^{a,[m]}, s_{t-1}^{r,[m]}]$  with a sample of the new state of the attributes  $s_t^{a,[m]} \sim p(s_t^a | a_t, s_{t-1}^a, s_{t-1}^r)$  and a sample of the new state of the relations  $s_t^{r,[m]} \sim p(s_t^r | a_t, s_t^a, s_{t-1}^r)$  (Lines 2 and 3). Subsequently the algorithm filters the temporary set of particles into the set  $\{S_t^{[m]}\}_{m=1}^M$  according to observation  $z_t$ . When an observation is acquired, the particles are weighted according to the sensor model in Eq. 8 (Line 4). Given our assumption that relations and activities are not directly observable, the sensor model takes into account only the part of the particles relative to the attributes but, since the particles are composed of three parts, these are also weighted. Weights are normalized (Line 5) and the set of particles for the next iteration is extracted using importance sampling (Line 6). The algorithm's consistency has been demonstrated in a way similar to the one used by Thrun, Burgard and Fox (2005) and reported in the Appendix.

## 5 EXPERIMENTS

We conduct experiments on two domains. Using the first, we examine our hypothesis that considering the relations between objects simultaneously improves tracking and activity recognition. Using the second domain, we show how the RPF can successfully resolve data association ambiguities while tracking objects in the presence of occlusion.

### 5.1 Sea Navigation

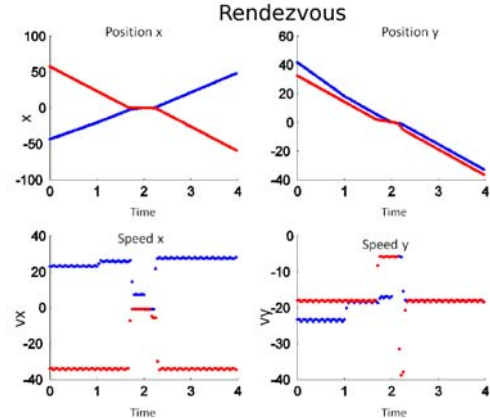


Figure 2: Example of a rendezvous: positions and speed for the two ships (blue and red).

We evaluate the RPF algorithm on the data set provided for the Intelligent Systems Challenge 2008-2009<sup>2</sup>. The data set includes 40 sequences. Each sequence comprises the tracks

<sup>2</sup>[www.intelligent-system-challenge.ca/home/index/html](http://www.intelligent-system-challenge.ca/home/index/html)

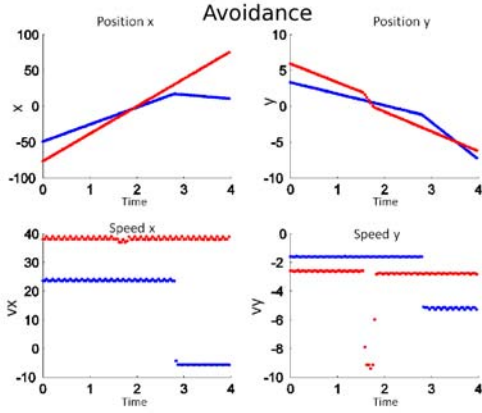


Figure 3: An example of an avoidance activity. For each ship position and speed are shown.

(i.e., series of 2D positions) of two ships participating in an activity. The ships are of three distinct *types*: *yacht*, *fisher* and *cargo ship*. For each type of ship, we are given prior knowledge of the average speed and the frequency of heading changes. In each sequence, at most one activity takes place, which can be either *rendezvous* or *avoidance*. The data set includes 19 rendezvous sequences and 21 avoidance sequences. Both activities can be described in terms of two ships approaching each other and then going apart. In an avoidance activity, only one of the two ships (following the rules of the sea) changes its speed to avoid the other. During a rendezvous, once the ships are close, they remain close with speeds near zero to exchange goods.

### 5.1.1 Settings

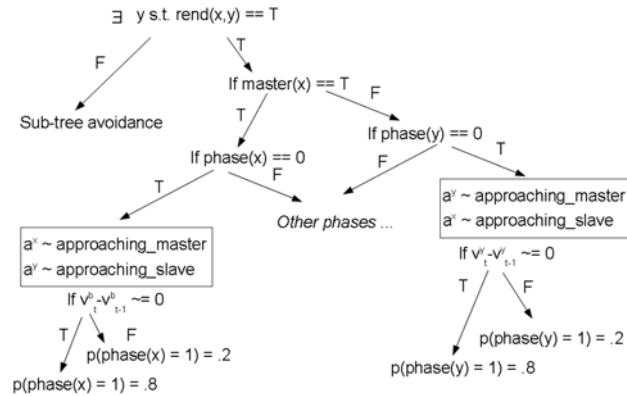


Figure 4: Relational transition model. At each time step, for each object it computes the next state and the future phase and relation given the current object’s relation and phase.

Since the behaviors of the ships involved in a joint activity are correlated, representing the relations between the ships

is particularly useful for recognizing this activity. For the rendezvous activity, we hypothesize the existence of a latent “master-slave” relation. According to our hypothesis, when changes in speed and direction occur, the ship with the *slave* role mirrors the behavior of the *master* ship with some delay (Fig. 2). For the avoidance activity it is expected that one of the two ships stops to let the other pass. For simplicity, we refer to the corresponding roles for the ships as *stopper* and *passer*. These relations are modeled probabilistically: each particle makes a particular assumption about the activity, in either case it assigns each ship to one of the two roles available; Bayesian inference, implemented with our RPF algorithm, makes sure that the particle population converges to the true role.

We model the two activities using three discrete phases. In a rendezvous activity (Fig. 2), the two ships approach each other (phase 0), they stay close to exchange goods (phase 1) and they go apart (phase 2). For an avoidance activity (Fig. 3), in phase 1, one ship stops or proceeds slowly to let the other pass. The representation of the phases is strictly probabilistic: each particle makes a hypothesis about the current activity and phase and then predicts the next state using a transition probability parameterized by the activity, the phase and the role assigned to each ship.

We represent the *state of the attributes* with the position ( $p$ ), the velocity ( $v$ ) and the type ( $type$ ) of each target in the scene. The dynamical model for the attributes  $[p_t, v_t, type_t]$  has the following form:

$$p_{t+1} = p_t + v_t dt + \frac{1}{2} a(activity, phase, role) dt^2$$

$$v_{t+1} = v_t + a(activity, phase, role) dt$$

$$type_{t+1} = type_t$$

where  $a(activity, phase, role, type)$  is a random variable whose distribution depends on the known type of the ship and the beliefs of the particle concerning the current activity, the phase of this activity, and role of the ship. For example, in a rendezvous a yacht acting as master tends to have a smaller acceleration during phase 0, when it decides to approach, and a cargo ship acting as slave tends to have a larger acceleration because it has to reach the master. We learned the distribution of the variable  $a$  from the data. For the sensor model, we assume the ship type is perfectly known and the values for the other attributes ( $[p_t, v_t]$  for each ship) are selected according to a Gaussian distribution with a small standard deviation (0.5).

Each particle represents its beliefs about the state of the relational domain (positions, velocities, ship types, relations) and the activity (rendezvous or avoidance), together with the activity phase and the role of the ship. The relational transition model is encoded using a probabilistic tree whose nodes are formulas. The arguments to the formula are the variables in the state (Fig. 4, which omits the decision level for the object type for clarity of presenta-

tion). For instance, if a particle assumes that the ongoing activity is a rendezvous and the ships are approaching each other (phase 0), it will predict the motion of the master ship with an acceleration distributed accordingly to a model called *approaching-master* where the acceleration is typically small and directed towards the other ship (Fig. 4). Conversely, the motion of the ship that is assumed to be the slave will be modeled with a model called *approaching-slave*, which typically has high acceleration. The particle will also predict the change of phase of the activity: if the speed of the ship believed to be the master approaches zero, with a certain probability (.8) there will be a change of phase from phase 1 to phase 2. The probabilities of transitions between phases are estimated from data.

Lacking other existing methods applicable to the problem we focus on, we compare our results with those obtained using a pair of HMMs (Rabiner, 1989), one for each activity. We define a hidden state of an HMM as a joint motion of the two ships involved in the activity and associate it with a probabilistic transitional model for the position and velocity of the ships. We separate the data into two sets (rendezvous and avoidance) and analyze velocity profiles of the ships. We cluster values of velocity in  $k$  classes per ship, which corresponds to  $k^2$  joint states (combination of velocity classes of the two ships). These joint states indirectly encode the three phases of each activity. We considered  $k = 2, 3, 4$ , corresponding to fixed numbers of joint states (4, 9 and 16, respectively) that are in the same order of magnitude of the number of phases of the activities. We learn a transition matrix for each activity given the joint states learned from the clustering. The emission matrix is given by the distance between clusters. It tells us how likely it is for a data point to be misclassified in a particular class. For the activity recognition task, for each sequence of positions of two ships, we compute the probability of it being a rendezvous as the activity whose HMM gives the higher probability.

We chose a system of HMMs for comparison because it is the most general approach in the activity recognition literature and to our knowledge it is one of the few online approaches. Moreover, the use of discrete states makes it possible to describe joint states even though relations are ignored. To our knowledge, no previous method is capable of taking relations into account, so the comparison with HMMs is reasonable.

### 5.1.2 Results

We ran the experiments on each of the 40 sequences in the data set. The first two rows of Table 1 compare the performance (in terms of precision and recall) of the RPF for the recognition of the “rendezvous” activity, with the performance of a system of HMMs modeling the activities with 4, 9 or 16 states. The last row of Table 1 compares the average tracking error ( $E$ ) for RPF and a conventional PF that

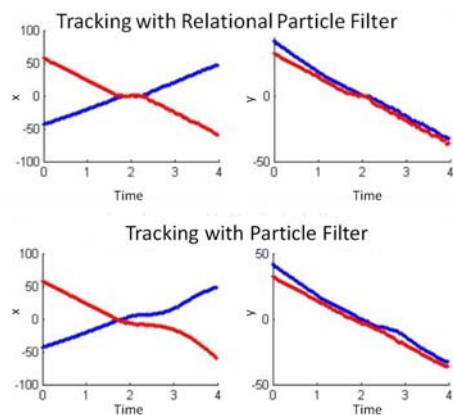


Figure 5: Tracking ships for the rendezvous activity shown in Fig. 2 with the RPF and PF algorithms.

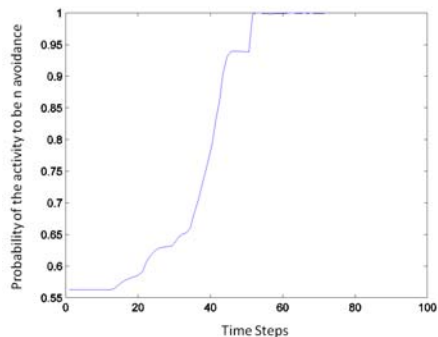


Figure 6: The average ratio of correctly recognized avoidance activities over time.

does not account for relations. The PF algorithm has been implemented in the exact same way as the RPF algorithm except that the random variable  $a$  depends on only the type of the ships and not on the hypothesized activity and role of the object (because this information is included in the relational model but not in the standard one). Each algorithm (PF and RPF) uses only 100 particles.

As shown in Table 1, the HMM with 9 states classifies almost all sequences as rendezvous, giving a high false negative rate and thus a low precision. The HMM with 16 states has relatively good performance, though lower than the performance of the RPF algorithm, using RPF for activity recognition results in a lower false alarm rate. Moreover, our method improves the tracking performance compared with a PF algorithm. We verified the importance of the master-slave relation: we compare the tracking error of the RPF method to the error computed considering only those tracks for which the role of the ships are correctly recognized: when the role of the ships are successfully recognized, the tracking error of our method is significantly lower (0.082 vs 0.154). In Fig. 5 a rendezvous tracked



	RPF	PF	HMM(4)	HMM(9)	HMM(16)
Precision	1		0.48	0.58	0.81
Recall	0.74		1	0.95	0.62
F1	0.85		0.64	0.72	0.71
$E$	0.15	1.68			

Table 1: Precision and recall for the activity recognition task of the RPF algorithm and an HMM algorithm. Tracking error of the RPF and PF algorithms compared (both PF use 100 particles).

with our RPF algorithm is compared with the same pair of tracks tracked with a PF algorithm (using 100 particles). Our method outperforms the standard tracking algorithm.

RPF supports online activity recognition. Fig. 6 shows the average belief of the avoidance over time. In just 40 time steps, our RPF algorithm selects the correct activity with a probability of more than 70%.

## 5.2 Tracking in the Presence of Occlusion



Figure 7: A video frame of the problem of Section 5.2.

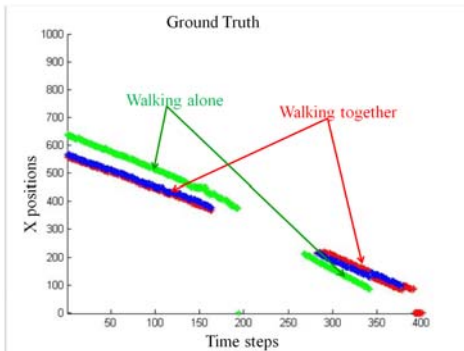


Figure 8: The real trajectories of the three people (ground truth).

The tracking ability of RPF is evaluated on the problem of tracking three people in an occlusion setting: two people are walking together in front of a third person and enter a “blind spot” (Fig. 7) but when the group exits the

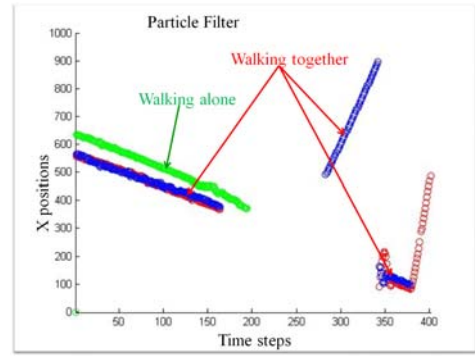


Figure 9: Multi-target tracking with a standard PF.

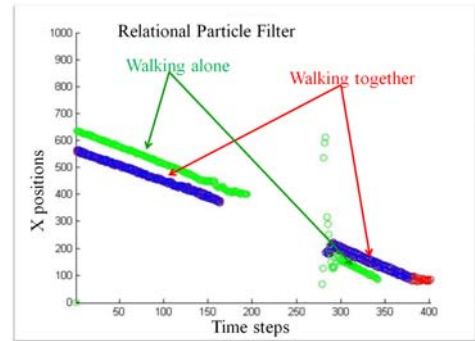


Figure 10: Multi-target tracking with our RPF.

“blind spot” the order of the people has changed and the lone walker is first. This experiment is challenging because a naive tracker will likely incorrectly associate people to tracks when they exit the “blind spot”.

In our settings we assume that people who are walking together are most likely to continue to walk together and are likely to have similar motion. We implement this assumption in the relational transition model used by our RPF algorithm. In these settings, the activity is that of walking together and the state of the attributes is given by the position and velocity of the three people in the scene ( $s_t = [p_t(1), v_t(1), p_t(2), v_t(2), p_t(3), v_t(3), ]$ ). We assume a person walking alone will maintain a constant velocity during time plus a certain random variable  $a$  whose distribution is learned from data:

$$p_{t+1}(i) = p_t(i) + v_t(i)dt + \frac{1}{2}a dt^2$$

$$v_{t+1}(i) = v_t(i) + a dt.$$

At the beginning each person has a certain probability (50%) of being walking together with one of the other persons. If two persons are hypothesized to be walking together (i.e., if a particle assumes they are walking together) their positions are predicted to be close. We first calculate the midpoint  $\mu$  of the two persons (i.e., the mean point of



the segment that links their position) and predict the position of this point with a velocity that is the average of the two persons' velocities plus a random variable:

$$\begin{aligned} p_t(\mu) &= \frac{1}{2}(p_t(i) + p_t(j)) \\ v_t(\mu) &= \frac{1}{2}(v_t(i) + v_t(j)) \\ p_{t+1}(\mu) &= p_t(\mu) + v_t(\mu)dt + \frac{1}{2}a dt^2. \end{aligned}$$

The distance between the two persons at time  $t$  is  $d_t(i, j) = \text{abs}(p_t(i) - p_t(j))$ . A random variable  $\delta$  represents how the distance between the persons walking together changes during time and it is learned from data. We compute the velocity of the two persons:

$$\begin{aligned} p_{t+1}(i) &= p_{t+1}(\mu) + d_t(i, j) + \delta \\ p_{t+1}(j) &= p_{t+1}(\mu) + d_t(i, j) + \delta \\ v_{t+1}(i) &= (p_{t+1}(i) - p_t(i))/dt \\ v_{t+1}(j) &= (p_{t+1}(j) - p_t(j))/dt. \end{aligned}$$

To solve the problem of data association, we use an approach similar to the *joint probabilistic data association* explained by Bar-Shalom (1987). At each time step, each particle is weighted according to the joint probability  $p(z_t | s_t^a)$  for each possible permutation of the objects in the domain<sup>3</sup>. If at time step  $t$  we observe the objects  $\alpha, \beta$  and  $\gamma$ , the weight  $\omega_t^{[m]}$  of the  $m$ -th particle  $S_t^{[m]}$  is computed as:

$$\begin{aligned} \omega_t^{[m]} &= p([\alpha, \beta, \gamma] | S_t^{[m]}) + p([\alpha, \gamma, \beta] | S_t^{[m]}) \\ &+ p([\beta, \gamma, \alpha] | S_t^{[m]}) + p([\beta, \alpha, \gamma] | S_t^{[m]}) \\ &+ p([\gamma, \alpha, \beta] | S_t^{[m]}) + p([\gamma, \beta, \alpha] | S_t^{[m]}). \end{aligned} \quad (10)$$

We compare the performance of our RPF with that of a standard PF that predicts the state of all the persons in the scene with a transition model that does not take into account relation (it has been implemented with the same model used for the walking alone activity). The graphs in Figs. 9 and 10 show that the RPF algorithm is better than the standard PF. After the first (lone) person exits the "blind spot" both methods identify him with one of the persons walking together. However, while RPF is able to quickly correct this assumption (the majority of the 1000 particles used converges to the correct association as soon as the correlated behavior of walking together is detected again after the blind spot) the PF, using the same number of particles, cannot recover the correct association and gives completely wrong tracks. Overall, the average tracking error for the RPF algorithm is 1.0955, 0.9595 and 2.5145 for the first,

<sup>3</sup>As sensor model we use a Gaussian distribution centered on the state of the attributes with standard deviation 0.4.

second and third person, respectively, and for the PF algorithm is 1.1780, 1.0029 and 2.5674 with a standard deviations over 100 runs of 0.2256, 0.1931 and 0.3152 (RPF) and 0.2266, 0.1931 and 0.3153 (PF).

## 6 DISCUSSION AND FUTURE WORK

This paper shows how the explicit recognition of relations between interacting objects can improve the understanding of their dynamic behavior. We use RDBNs to represent probabilistic dependencies between objects in the context of online active recognition and multi-target tracking. Our experiments show that RDBNs have significant advantages over methods that do not use relations.

While relations provide rich domain knowledge, they also increase the dimensionality of the state space. Thus, as the number of relations increases, so must the number of particles used. Accordingly, one should be careful not to introduce uninformative relations, since RPF, like all Monte Carlo methods, can require a prohibitive number of particles to perform inference in high-dimensional state spaces.

In future work RDBN and RPF could be applied to more challenging domains with larger state spaces. While in this work we have manually specified the dynamical models, it is clear that this becomes tedious for larger models. Accordingly, a natural direction for future work is the automatic learning of such relational models. A second direction for future work is the extension of the approach to first-order theories. There do exist papers with first order logic and probabilistic reasoning where objects' attributes and relations do not vary with time (e.g., see Sec. 2). The extension of RDBNs to first-order theories introduces several challenges, especially the development of an effective approach to inference without first grounding all the first order clauses that are true in the domain.

## APPENDIX

In what follows we sketch the derivation of the RPF algorithm. Following the formulation of Thrun, Burgard and Fox (2005), we redefine a particle at time step  $t$  as a state sequence up to time  $t$ :

$$[y_{0:t}^{[m]}, x_{0:t}^{a,[m]}, x_{0:t}^{r,[m]}] \equiv [[y_0^{[m]}, x_0^{a,[m]}, x_0^{r,[m]}], \dots, [y_t^{[m]}, x_t^{a,[m]}, x_t^{r,[m]}]].$$

The RPF algorithm needs to be modified accordingly: We append to the particle  $[y_t^{[m]}, x_t^{a,[m]}, x_t^{r,[m]}]$  the sequence of state samples from which it was generated  $[y_{0:t-1}^{[m]}, x_{0:t-1}^{a,[m]}, x_{0:t-1}^{r,[m]}]$ . Then, the RPF algorithm calculates the belief over all state sequences:  $\text{bel}(S_{0:t}) = p(S_{0:t} | z_{1:t})$ . This equivalent formulation of the problem is used to derive the RPF algorithm in Algorithm 1.

$bel(S_{0:t})$  is obtained by the following substitutions (where the absence of the integral is the result of maintaining the full posterior rather than the marginal filtering distribution):

$$\begin{aligned}
bel(S_{0:t}) &= p(S_{0:t}|z_{1:t}) \\
&\stackrel{Bayes}{=} \alpha p(z_t|S_{0:t}, z_{1:t-1})p(S_{0:t}|z_{1:t-1}) \\
&\stackrel{Markov}{=} \alpha p(z_t|S_t)p(S_{0:t}|z_{1:t-1}) \\
&= \alpha p(z_t|S_t)p(S_t|S_{0:t-1}, z_{1:t-1}) \\
&\quad p(S_{0:t-1}|z_{1:t-1}) \\
&\stackrel{Markov}{=} \alpha p(z_t|S_t)p(S_t|S_{t-1})p(S_{0:t-1}|z_{1:t-1}) \\
&\stackrel{Eq. 7}{=} \alpha p(z_t|S_t)p(a_t|S_{t-1})p(s_t^a|a_t, s_{t-1}) \\
&\quad p(s_t^r|a_t, s_t^a, s_{t-1}^r)p(S_{0:t-1}|z_{1:t-1})
\end{aligned}$$

where  $\alpha$  is a normalizing factor. The derivation is now carried out by induction. To verify the initial condition, we simply assume that our first particle set is obtained by sampling the prior  $p(S_0)$ . Let us assume that the particle set at time  $t - 1$  is distributed according to  $bel(S_{0:t-1})$ . For the  $m$ -th particle  $[y_{0:t-1}^{[m]}, x_{0:t-1}^{a,[m]}, x_{0:t-1}^{r,[m]}]$  in the input set, the sample  $[y_t^{[m]}, x_{t-1}^{a,[m]}, x_{t-1}^{r,[m]}]$  is generated from the proposal distribution:

$$p(a_t|S_{t-1})bel(S_{0:t-1}),$$

the sample  $[y_t^{[m]}, x_t^{a,[m]}, x_{t-1}^{r,[m]}]$  is generated according to:

$$p(s_t^a|a_t, s_{t-1})p(a_t|S_{t-1})bel(S_{0:t-1}),$$

with  $a_t = y_t^{[m]}$ , and the sample  $[y_t^{[m]}, x_t^{a,[m]}, x_{t-1}^{r,[m]}]$  is generated according to the proposal distribution:

$$p(s_t^r|a_t, s_t^a, s_{t-1}^r)p(s_t^a|a_t, s_{t-1})p(a_t|S_{t-1})bel(S_{0:t-1})$$

with  $a_t = y_t^{[m]}$  and  $s_t^a = x_t^{a,[m]}$ .

To obtain a properly weighted sample set, the importance weights are given by the ratio of the target and proposal distributions, where target distribution =  $\alpha p(z_t|S_t)p(s_t^r|s_{t-1}^r, s_t^a)p(s_t^a|a_t, s_{t-1})p(a_t|S_{t-1})bel(S_{0:t-1})$  (for some constant  $\alpha$ ) and proposal distribution =  $p(s_t^r|s_{t-1}^r)p(s_t^a|a_t, s_{t-1})p(a_t|S_{t-1})bel(S_{0:t-1})$ . In this way  $\omega_t^{[m]} = \alpha p(z_t|S_t)$ .

The constant  $\alpha$  plays no role since the resampling takes place with probability proportional to the importance weights. The resulting particles are distributed according to the product of the proposal and the importance weights:

$$p(s_t^r|s_{t-1}^r, s_t^a)p(s_t^a|a_t, s_{t-1})p(a_t|S_{t-1})bel(S_{0:t-1})p(z_t|S_t),$$

that is  $bel(S_{0:t})$ .

The consistency of the RPF algorithm follows from the observation that if  $[y_t^{[m]}, x_{0:t}^{[m],a}, x_{0:t}^{[m],r}]$  is distributed according to  $bel(S_{0:t})$ , then the augmented relational state sample  $[y_t^{[m]}, x_t^{[m],a}, x_t^{[m],r}]$  is distributed according to  $bel(S_t)$ .

## References

- Arulampalam, S., Maskell, S., & Gordon, N. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. Signal Processing*, 50, 174–188.
- Bar-Shalom, Y. (1987). *Tracking and data association*. Academic Press Professional, Inc., San Diego, CA, USA.
- Black, M., & Fleet, D. J. (2000). Probabilistic detection and tracking of motion boundaries. *International Journal of Computer Vision*, 38(3), 231 – 245.
- Elidan, G., Heitz, G., & Koller, D. (2006). Learning object shape: From drawings to images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Friedman, N., Getoor, L., Koller, D., & Pfeffer, A. (1999). Learning probabilistic relational models. In Dean, T. (Ed.), *IJ-CAI*, pp. 1300–1309. Morgan Kaufmann.
- Getoor, L., Friedman, N., Koller, D., & Taskar, B. (2002). Learning probabilistic models of link structure. *Journal of Machine Learning Research*, 3, 679–707.
- Hoiem, D., Efros, A. A., & Hebert, M. (2006). Putting objects in perspective. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, Vol. 2, pp. 2137 – 2144.
- Isard, M., & Blake, A. (1998). A mixed-state condensation tracker with automatic model-switching. In *ICCV*, pp. 107–112.
- Ivanov, Y. A., & Bobick, A. F. (2000). Recognition of visual activities and interactions by stochastic parsing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8), 852–872.
- Jaeger, M. (1997). Relational bayesian networks. In Geiger, D., & Shenoy, P. P. (Eds.), *UAI*, pp. 266–273. Morgan Kaufmann.
- Manfredotti, C. (2009). *Modeling and Inference with Relational Dynamic Bayesian Networks*. Ph.D. thesis, Computer Science Department, University of Milano-Bicocca, Italy.
- Manfredotti, C. E., & Messina, E. (2009). Relational dynamic bayesian networks to improve multi-target tracking. In Blanc-Talon, J., Philips, W., Popescu, D. C., & Scheunders, P. (Eds.), *ACIVS*, Vol. 5807 of *Lecture Notes in Computer Science*, pp. 528–539. Springer.
- Milch, B., & Russell, S. J. (2006). General-purpose mcmc inference over relational structures. In *UAI*. AUAU Press.
- Poon, H., Domingos, P., & Sumner, M. (2008). A general method for reducing the complexity of relational inference and its application to mcmc. In Fox, D., & Gomes, C. P. (Eds.), *AAAI*, pp. 1075–1080. AAAI Press.
- Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pp. 257–286.
- Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.
- Tran, S. D., & Davis, L. S. (2008). Event modeling and recognition using markov logic networks. In Forsyth, D. A., Torr, P. H. S., & Zisserman, A. (Eds.), *ECCV (2)*, Vol. 5303 of *Lecture Notes in Computer Science*, pp. 610–623. Springer.