

# Model-based Multi-agent Tracking

## Abstract

## Introduction

*Multi-agent tracking* is the task of locating multiple moving objects (the agents) over time. In such task, the assumption of having fully observable trajectories of the moving objects means that the sensors are able to capture every state change at every instant, which typically is unrealistic. Normally obtaining sensor feedback (or the processing of the sensor readings) is associated with a given sampling frequency that misses intermediate data between two subsequent sensor readings.

This work exploits (1), a movement model of the agents to track and (2), an off-the-shelf classical planner, to produce reliable multi-agent tracking (even when there is a significant amount of missing sensor data). The paper formalizes the *Model-based Multi-Agent Tracking* task and shows that this task is addressable with an off-the-shelf classical planner. Last but not least the paper shows the applicability of this approach for tracking ants.

## Background

This section defines the planning model and the observation model that we follow in this work.

### The Classical Planning Model

We define a *classical planning* problem as a tuple  $P = \langle X, I, G, A \rangle$ :

- $X$ , is the set of *state variables* such that each variable  $x \in X$  has an associated finite domain  $D(x)$  defining the set of possible values that the variable can take. The set of possible states  $S$  in a classical planning problem  $P$  is then given by the *Cartesian product* of the size of the domain variables.
- $I$  is the *initial state*, a full assignment to the variables in  $X$  that is compatible with their respective domains.
- $G$  a conjunction of *goal conditions* where each  $g \in G$  expresses as different Boolean condition over the state variables in  $X$ , that is  $g : S \rightarrow \{0, 1\}$ .

- $A$  is the set of actions. The semantics of actions  $a \in A$  is expressed with two functions:

- The *applicability function*  $\alpha : A \times S \rightarrow \{0, 1\}$  which determines if a given action  $a \in A$  is applicable in a given state  $s \in S$ .
- The *transition function*  $\theta : A \times S \rightarrow S$  determining which is the successor state  $\theta(s, a)$  that results after applying action  $a \in A$  at the current state  $s \in S$ .

A *plan* for a classical planning problem  $P$  is an action sequence  $\pi = \langle a_1, \dots, a_n \rangle$  that induces the *state trajectory*  $s = \langle s_0, s_1, \dots, s_n \rangle$  such that  $s_0 = I$  and, for each  $1 \leq i \leq n$ ,  $a_i$  is applicable in  $s_{i-1}$  and generates the successor state  $s_i = \theta(s_{i-1}, a_i)$ . The *plan length* is denoted with  $|\pi| = n$ . A plan  $\pi$  *solves*  $P$  iff  $G \subseteq s_n$ , i.e., if all the goal conditions are satisfied at the last state reached after following the application of the plan  $\pi$  in the initial state  $I$ . A solution plan for  $P$  is *optimal* if it has minimum length.

### The Observation Model

Given a classical planning problem  $P = \langle X, I, G, A \rangle$ , we define  $\omega_s = \text{obs}(s)$  as the observation of a state  $s \in S$  which is as a partial assignment of the state variables in  $X$ . In this work we assume that state observations are noiseless, this means the variable values given by  $\omega_s$  are correct while the value of some variables in  $X$  may be unknown.

Now, we define the observation of the execution of a plan  $\pi$  that solves  $P$  as a sequence  $\Omega_\pi = \langle \text{obs}(s_0), \dots, \text{obs}(s_m) \rangle$  of *partially-observed* states. Because of the partial observability  $0 \leq |\Omega_\pi| \leq |\pi| + 1$  and hence, the transitions between two consecutive observed states may involve the execution of more than a single action. Formally  $\theta(s_i, \langle a_1, \dots, a_k \rangle) = s_{i+k}$ , where  $k \geq 1$  is unknown and unbound. This means that having  $\Omega_\pi$  does not implies knowing the actual length of  $\pi$ .

We say that a given plan  $\pi$  is *compliant* with a given sequence  $\Omega_\pi = \langle \text{obs}(s_0), \dots, \text{obs}(s_m) \rangle$  of *partially-observed* states iff  $\Omega_\pi$  is the same sequence of states traversed by  $\pi$  but:

1. with certain states omitted and/or,
2. the value of certain fluents omitted in that states, i.e.  $|\omega_{s_i}| \leq |X|$  for every  $0 \leq i \leq m$ .

t0	t1	t2	t3
X Y	X Y	X Y	X Y
33 33	71 23	30 54	43 10
72 22	95 11	93 89	59 36
19 93	86 05	21 18	18 38
58 65	06 60	67 94	78 60
83 61	75 76	23 19	68 54
16 53	17 98	41 59	59 96
02 97	89 11	51 84	86 08
75 23		79 69	87 60
74 01		36 13	
66 49			

Figure 1: 2D coordinates for ten agents in four subsequent time-stamps (at t1, t2 and t3 some readings are missing).

## Model-based Multi-Agent Tracking

*Model-based Multi-Agent Tracking* is the task of locating  $k$  moving objects by building the most plausible explanation of their trajectories that is compliant with (1), a model of the objects movement and (2), a given set of observations.

Formally the inputs to the *Model-based k-Agent Tracking* task are  $\mathcal{T} = \langle \Lambda, \mathcal{M}, \Omega \rangle$ , where:

- $\Lambda = \{\lambda_1, \dots, \lambda_k\}$ , is a set of labels to uniquely identify each of the  $k$ -agents.
- $\Omega = \langle obs(s_0), \dots, obs(s_m) \rangle$ , is a sequence of state observations. This observations express the configurations of the different agents to track, in all this observations agents are unlabeled.
- $\mathcal{M}$  the agent model. This model represents the space of the possible configurations of an agent. The agent model is implicitly represented using two elements:
  - $X_{\mathcal{M}}$ , a finite set of variables with their corresponding domain.
  - The functions defining the possible updates of the agent variables between to subsequent time-stamps  $t$  and  $t + 1$ .

The output to the *Model-based Multi-Agent Tracking* task is, for each agent, a probability distribution  $P(\lambda|\Omega, \mathcal{M})$  of the possible labels that the agent can have given the observations and the agent models.

To illustrate this, Figure 1 shows an example of four observations of the 2D coordinates of ten agents (at time-stamps t1, t2 and t3 the coordinates of some of the ten agents are missing because they are unknown). Note that observations in the same row do not necessary refer to the same agent (the agents identity is unknown). This task requires ten labels to uniquely identify the different agents and the agent model is given by to discrete variables  $X, Y$  that models the current position of the agent and the following action model  $\{x_{t+1} = x_t + 1, x_{t+1} = x_t - 1, y_{t+1} = y_t + 1, y_{t+1} = y_t - 1\}$  that represent the possible updates of the  $X, Y$  between to subsequent time-stamps.

## Explaining observations with Classical Planning

Given an agent model  $\mathcal{M}$  and a set of observations  $\Omega = \langle obs(s_0), \dots, obs(s_m) \rangle$ , we can build a classical planning

```
0: dec-x(a1,114,113) dec-x(a2,110,19) dec-y(a3,111,110)
dec-y(a4,113,112) dec-y(a5,18,17) inc-x(a6,15,16)
inc-x(a7,16,17) inc-x(a8,12,13) inc-x(a9,13,14)
inc-x(a10,15,16)
```

```
1: dec-x(a1,114,113) dec-x(a2,110,19) dec-y(a3,111,110)
dec-y(a4,113,112) dec-y(a5,18,17) inc-x(a6,15,16)
inc-x(a7,16,17) inc-x(a8,12,13) inc-x(a9,13,14)
inc-x(a10,15,16)
```

```
2: dec-x(a1,114,113) dec-x(a2,110,19) dec-y(a3,111,110)
dec-y(a4,113,112) dec-y(a5,18,17) inc-x(a6,15,16)
inc-x(a7,16,17) inc-x(a8,12,13) inc-x(a9,13,14)
inc-x(a10,15,16)
```

Figure 2: Plan simulating the movement of ten agents in a grid.

problem  $P = \langle X, I, G, A \rangle$  such that a solution plan to this problem is an explanation of the input state observations and an optimal solution plan to this problem is the smallest explanation for given input observations and agent model.

The classical planning problem  $P = \langle X, I, G, A \rangle$  for explaining a set of observations  $\Omega = \langle obs(s_0), \dots, obs(s_m) \rangle$  with an agent model  $\mathcal{M}$  is built as follows:

1. The set of state variables  $X$  is build defining a set of variables  $X_{\mathcal{M}}$  for each agent in the observations. In addition an extra state variable  $X_t$  with discrete domain  $[0, m]$  is added to keep track of the current time-stamp.
2. The initial state  $I$  is a full assignment of the agent variables for each of the agents (note that in t0 the 2D coordinates of all agents are known).
3. The set of goals  $G$  is a conjunction of the conditions forcing the plan to generate the observations at their precise times-stamp. For instance, for the time stamp  $t1$  we have 7 conditions over the  $X$  variables of the agents and another 7 over the  $Y$  variables.
4. The set of actions models the functions defining the possible updates of the agent variables between to subsequent time-stamps  $t$  and  $t + 1$ .

Figure 2 shows a three steps plan that is *compliant* with the agent model described in the previous example and the observations given in Figure 1.

## Model-based Multi-agent Tracking with Classical Planning

Here we show that the probability distribution  $P(\lambda|\Omega, \mathcal{M})$  that solves a *Model-based Multi-Agent Tracking* can be estimated using an off-the-shelf classical planning problem.

## Evaluation: Ants tracking

### Conclusions

### References