Model-based Multi-agent Tracking

Abstract

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 appliability 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 *cartession 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 $\pi = \langle a_1, \ldots, a_n \rangle$ that induces the *state trajectory* $s = \langle s_0, s_1, \ldots, s_n \rangle$ 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 π *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 π 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 $\omega_s = obs(s)$ as the observation of a state $s \in S$ which is as a partial assigment of the state variables in X. In this work we assume that state observations are noiseless, this means the variable values given by ω_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 π that solves P as a sequence $\Omega_\pi = \langle obs(s_0), \ldots, obs(s_m) \rangle$ 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 $\theta(s_i, \langle a_1, \ldots, a_k \rangle) = s_{i+1}$, where $k \geq 1$ is unknown and unbound. This means that having Ω_π does not implies knowing the actual length of π.

We say that a given plan π is *compliant* with a given sequence $\Omega_\pi = \langle obs(s_0), \ldots, obs(s_m) \rangle$ of *partially-observed* states iff Ω_π is the same sequence of states traversed by π 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.

Model-based Multi-Agent Tracking

Model-based Multi-Agent Tracking is the task of locating k moving objects by building the most possible explanation of their trajectories that is compliant with a model of the objects movement and a set of observations.

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

- A set of labels $\Lambda = \{\lambda_1, \ldots, \lambda_k\}$ to uniquely identify each of the k-agents.

- A sequence of state observations $\Omega = \langle obs(s_0), \ldots, obs(s_m) \rangle$. This observations express the configurations of the different agents to track, in all this observations agents are unlabeled. To illustrate this, Figure 1 shows an example of four observations of the 2D coordinates of ten agents (in time-stamps t1, t2 and t3 the coordinates of some of the ten agents is unknown). Note that observations in the same row do not neccesaary refer to the same agent.

- A movement model $\mathcal{M}$. This model expresses the set of possible configuration changes of an agent between to subsequent time-stamps t and $t + 1$. An example of a possible movement for each of the agents to change its configuration between to subsequent time-stamps could be $\mathcal{M} = \{x_{t+1} = x_t + 1, x_{t+1} = x_t - 1, y_{t+1} = y_t + 1, y_{t+1} = y_t - 1\}$.

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.

Tracking as 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

```
0: move-dec-x(ant1,l1,l0) move-dec-x(ant3,l15,l14) move-dec-x(ant8,l10

1: move-dec-x(ant3,l14,l13) move-dec-x(ant9,l10,l9) move-dec-y(ant1,l1

2: move-dec-x(ant3,l13,l12) move-dec-y(ant1,l10,l9) move-dec-y(ant3,l1

3: move-dec-x(ant3,l12,l11) move-dec-y(ant3,l11,l10) move-inc-x(ant10,

4: move-dec-x(ant2,l9,l8) move-dec-x(ant3,l11,l10) move-dec-x(ant9,l9,

5: move-dec-x(ant3,l10,l9) move-dec-x(ant9,l8,l7) move-dec-y(ant2,l9,l

6: move-dec-x(ant7,l9,l8) move-dec-x(ant8,l9,l8) move-dec-y(ant10,l9,l

7: move-dec-x(ant3,l9,l8) move-dec-x(ant6,l9,l8) move-dec-y(ant6,l9,l8

8: move-dec-x(ant3,l8,l7) move-dec-x(ant5,l9,l8) move-dec-x(ant8,l8,l7

9: move-dec-x(ant6,l8,l7) move-dec-x(ant7,l9,l8) move-dec-x(ant8,l7,l6

10: move-dec-x(ant3,l7,l6) move-dec-x(ant6,l7,l6) move-dec-x(ant7,l8,l

11: move-dec-x(ant3,l6,l5) move-dec-x(ant6,l6,l5) move-dec-x(ant7,l7,l

12: move-dec-x(ant10,l9,l8) move-dec-x(ant3,l5,l4) move-dec-x(ant6,l5,

13: move-dec-x(ant10,l8,l7) move-dec-x(ant3,l4,l3) move-dec-x(ant6,l4,

14: move-dec-x(ant10,l7,l6) move-dec-x(ant2,l9,l8) move-dec-x(ant3,l3,
```

Figure 2: Plan simulating the movement of ten agents in a 15×15 grid.