Assignment 2: Multi-camera Object Tracking

CSE5022 Advanced Multi-Agent Systems

DDL: 23:59, April 27, 2025

1 Overview

In this assignment, we will examine a Multi-agent System (MAS) where multiple fixed cameras statically placed in an environment cooperate to track a set of moving target objects. This Multi-camera Object Tracking scenario originates from the *Distributed Smart Camera Network* use case proposed in a paper [1], where the authors described a *socio-economic* inspired approach to effectively handle the tracking of multiple moving target objects. In short, we will use the open-source Repast Simphony Simulation Toolkit[1] to implement the scenario and conduct some experiments to analyze the tracking performance.

2 Objectives

This assignment aims to help the students get familiar with different types of communications among the agents.

1. Direct communication: the agents directly send messages to one another in a blocking/asynchronous fashion.
2. Indirect communication: the agents leave marks for other agents in the environment, and identify marks from other agents in the environment.

Therefore, try to put more focus on how to implement the communication logic in MAS.

3 About the Paper

This assignment essentially implements the paper [1], so it might be necessary to provide a brief introduction here. If you are already familiar with the paper, you can safely skip this section. Basically, there are multiple fixed cameras as well as multiple moving target objects in the environment. The ultimate goal for the camera network is to keep track of these target objects with the best effort. Each camera is placed in a fixed location and cannot move, but it can rotate a certain angle. In 2D space, its Field of View (FOV) is

[1] https://repast.github.io/

represented as a triangle. Target objects can only be tracked when they are within the FOV of the camera. Otherwise, they need to be handled by other cameras instead. This process is termed handover, where the camera transfers the ownership of the target object to another camera. Figure 1 shows the camera handover algorithm from the paper.

ALGORITHM 1: The Camera Handover Algorithm

(1) *Object trading of camera i*
  (a) Advertise owned objects to each other camera x with probability $P(i, x)$.
  (b) For each received advertised object j, respond with a bid at value $u_i(j)$ if this is greater than zero.
  (c) Accept received bids for each object k for which $u_i(k)$ is less than the highest received bid. For each accepted bid:
    i. Remove k from O_i.
    ii. Respond to the camera making the highest bid, informing it of the required payment, the value of the second highest received bid.
    iii. Increment the utility of the camera by the value of the second highest bid.
  (d) For each object l for which the bid sent was accepted, add l to O_i and deduct the payment amount from the utility of the camera.
(2) *Vision graph update of camera i*: Update τ_{ix} for all x according to Equation (3).
(3) *Tracking decisions of camera i*: Select which objects in O_i to track in order to maximise $U_i(O_i)$.
(4) Repeat at regular intervals.

Figure 1: Algorithm 1 from the paper.

Essentially, the handover process is based on Vickrey Auction described in [2]. When a camera i decides to hand over one of its owned target objects $k \in O_i$, it will notify all its neighboring cameras $N_G(i)$ to bid for the object. Then, the camera will select the winner with the highest bid and charge the winner the second-highest bid. The bid from a camera i depends on its calculated utility $u_i(k) = c_k \cdot v_k \cdot \phi_i(k)$ to track object k. $c_k \in [0, 1]$ specifies the estimated tracking performance of the target object, which depends on factors like distance, orientation, and partial occlusion. $v_k \in [0, 1]$ specifies the visibility of the target object, which is determined by the distance and angle of the object to the camera. Due to resource limitations, sometimes a camera cannot track all its owned objects. $\phi_i(k) \in \{0, 1\}$ specifies whether the object is currently tracked by the camera.

In the camera network, neighborhood relationships are maintained in a vision graph $G = (V, E)$, where the nodes V are cameras and an edge $e_{i,x} \in E$ suggests camera i to notify camera x when a handover occurs. Initially, each camera might choose to notify all other cameras (i.e., $|V| - 1$ edges per camera). To reduce the communication overhead, this vision graph can by dynamically updated by specifying pheromone levels, which are originally introduced in Ant Colony Optimization (ACO). Equation 1 specifies how the pheromone level τ_{ix} of edge $e_{i,x}$ is calculated, where ρ represents the evaporation rate and Δ represents the intensity of pheromone to be placed.

$$\tau_{ix} = \begin{cases} (1-\rho) \cdot \tau_{ix} & \text{if no trade occurs on the edge} \\ (1-\rho) \cdot \tau_{ix} + \Delta & \text{if trade occurs on the edge.} \end{cases} \tag{1}$$

It can be inferred that when trades (i.e., handovers) never happen between two cameras, then the edges connecting the two in the vision graph will finally be removed. Furthermore, each camera i tends to approach the neighboring cameras $x \in N_G(i)$ with more historical trades. To this end, a probability value $P(i, x)$ is calculated.

Finally, each camera specifies its instantaneous utility $U_i(O_i, p, r) = \sum_{k \in O_i} u_i(k) - p + r$, where p and r represent the total payments made and received in trades, correspondingly. Each camera will select a subset of owned target objects to track, in order to maximize this utility value U_i.

4 Requirements

1. Initialize the cameras and target objects in 2D space. Since the FOV of a camera is a triangle, you might need to manually implement the logic to check whether a target object is within its FOV (rather than using grid search in our previous demos). Visualizing FOV is not compulsory, although it might be helpful for debugging and demonstration. You need to implement at least one of the scenarios depicted in Figure 2.

2. In the paper, there are two approaches to calculate $P(i, x)$. You need to select one for your experiments. Briefly analyze the expected effect (i.e., what should happen) of this approach in your report.

3. Design functions to calculate c_k and v_k. Explain your design in the report. You can also make the scenario more interesting by introducing the tracking priority of target objects (optional).

4. Implement Vickrey Auction for the handover process. The communication can be either blocking (recommended as it is simpler) or asynchronous. In the paper, the broadcast/notification can be active or passive. For this assignment, we will use passive notification (i.e., notify only when the target object is out of the camera's FOV).

5. Implement pheromone management of the vision graph. You can refer to the ACO demo for more information. Explain in your report how you select ρ and Δ.

6. Implement selective object tracking based on the camera's instantaneous utility U_i. Describe the selection logic in your report.

7. From your simulation experiments, are the target objects effectively tracked by the camera network? Demonstrate the overall tracking performance in your report to answer this question. You might want to create some charts in the simulation interface.

8. From your simulation experiments, as the vision graph evolves, do cameras prefer to trade with only a subset of their neighbors? To answer this question, you can define a pheromone level threshold to filter out the favorite neighbors per camera and demonstrate in your report the average number of favorite neighbors using a chart, or you can design your own approach. Note that the result might differ if you configure cameras with different capabilities and available resources.

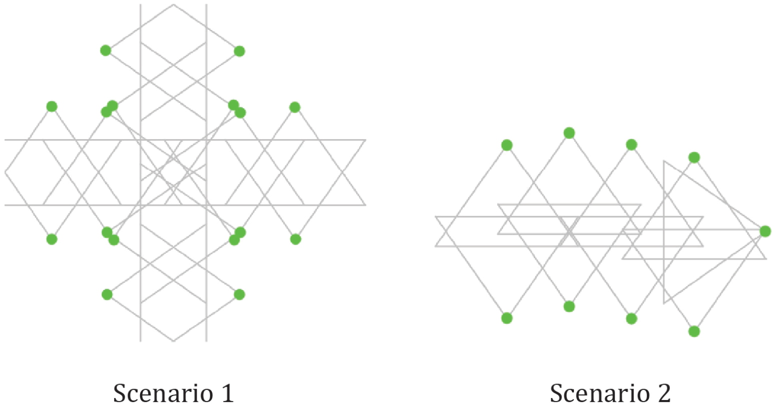

Figure 2: Illustrations of the simulation scenarios/environments.

5 What to Submit

1. **A report in PDF format**: Explain the main techniques proposed by the paper [1] based on your understanding and present any issues you find confusing in the paper. Then, describe how you implement the communications among agents. Finally, mention how you conduct the experiments and analyze the simulation results. Use screenshots for clear demonstrations and include as much detail as possible.

2. **Source code**. Describe in your report how to run simulations with your code.

Pack all files into `SID_NAME_A2.zip`, where `SID` is your student ID and `NAME` is your name (e.g., 11710106_ 张三 _A2.zip).

6 References

[1] Lukas Esterle et al. "Socio-economic vision graph generation and handover in distributed smart camera networks". In: ACM Transactions on Sensor Networks (TOSN) 10.2 (2014), pp. 1–24.

[2] Michael Wooldridge. An introduction to multiagent systems. John wiley & sons, 2009.