ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240630

CAMAOT: Channel-Aware Multi-Camera Active Object Tracking System

Maolong Yin[a], Bin Guo[a,*], Zhuo Sun[a], Lei Wu[a], Zhaotie Hao[a] and Zhiwen Yu[a]

[a]Northwestern Polytechnical University

Abstract. Multi-Camera Active Object Tracking is an attractive technique in the area of intelligent surveillance, where cameras share their observations via the wireless communication to collaboratively track the target. Due to the variability in wireless channel, the dynamic transmission delay between cameras significantly affects the collaboration performance, especially when the tracking is time-sensitive. In this paper, we propose a channel-aware multi-camera active object tracking (CAMAOT) system, to achieve the stable and improved tracking performance. Specifically, a communication decision module is designed in CAMAOT, where the cameras' communication graph and communication resource allocation adapt to the channels. Our experiments demonstrate that for time-varying channels, CAMAOT has a stable performance improvement over other systems, particularly when the communication resources are limited.

1 Introduction

With the rapid advancement of artificial intelligence technology, the multi-agent system has emerged. In the multi-agent system, multiple agents can collaboratively complete a specific task. Owing to its improved performance and stability, the multi-agent system is applied as a promising solution in several scenarios, such as smart manufacturing and intelligent surveillance. In the area of active object tracking (AOT), the collaboration of multiple rotatable cameras is proposed as a new paradigm. By sharing the information, the rotatable cameras collaborate to adjust their angles and improve the tracking performance of each camera.

For the initial studies on collaborative active object tracking, the coverage is a critical performance metric. In [4], the authors designed a collaborative active object tracking system to maximize the coverage, where each camera is regarded as an agent and its rotation is controlled by using Multi-Agent Reinforcement Learning (MARL)[30]. Based on that, the tracking accuracy is proposed as another performance metric. To maximize the tracking accuracy, a pose-assisted collaborative active object tracking system is proposed by [9]. In this system, the camera can predict the target's position, which is lost by the occlusion, from other cameras' sharing poses, thereby determining its rotation angle. While sharing pose information provides an efficient cooperation way, the pose information only indicates the target's current position. Due to the inevitable communication delay, it is difficult for the rotatable camera to follow the target in time, especially for the high-speed moving target. To address this issue, a collaborative active object tracking system called Effi-MAOT [29] was proposed. By sharing observation features, the cameras can predict the target's position from the current features and previous ones. This improves the prediction accuracy and the tracking performance for the high-speed moving target, especially when taking the communication delay into account. However, the aforementioned studies assume an ideal communication case, where the communication ability, e.g., the transmission rate, between any two cameras is homogeneous and remains constant. In practice, the time-varying channel results in heterogeneous and dynamic communication ability between any two cameras. This may cause intolerantly large delay for a pair of cameras, thereby deteriorating the overall tracking performance.

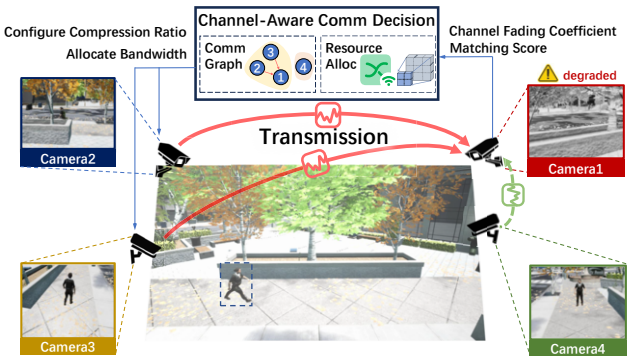


Figure 1: Illustration of Channel-Aware Multi-Camera Active Object Tracking system. The communication decision module of degraded cameras dynamically adjusts the communication graph and communication resource allocation based on channel conditions and matching scores, improving the tracking performance.

In this paper, we propose a channel-aware collaborative active object tracking system based on rotatable cameras, called CAMAOT. In the proposed system, the cameras share compressed features of their observations via wireless communication, when one camera's observation is degraded by the occlusion, as illustrated in Fig. 1. The wireless channel between any two cameras is time-varying fading. Based on the channels, each camera dynamically adjusts its communication graph, bandwidth allocation, and compression ratio to minimize the communication delays, thereby improving the object tracking performance. In order to verify the performance of the proposed system, we conduct extensive experiments in a high-fidelity virtual urban street environment. The widely used Rayleigh fading model is employed for wireless channels among the cameras. The experiments demonstrate that the proposed system significantly outperforms baseline systems, in terms of tracking performance and the amount of communication resource consumption.

* Corresponding Author. Email: guob@nwpu.edu.cn

2 Related Works

2.1 Active Object Tracking

AOT refers to the technique where a tracker automatically controls its motion based on the observation sequence and continuously track the target.

With the requirement of motion control adaptive to observations, the reinforcement learning (RL) based end-to-end AOT algorithms are proposed in [14, 32, 4, 9]. The authors in [14] adopted a ConvNet-Long Short-Term Memory (LSTM) function approximator to achieve the direct frame-to-action prediction. The adversarial reinforcement learning method was exploited to learn a robust tracker in [32], where the tracker and the target are two opponent agents and they can enhance each other during the competition. In [4], the authors extended the single-camera AOT to a multi-camera setting, where multiple cameras cooperatively track multiple targets and each camera's rotation is controlled by using MARL. The authors in [9] designed a pose-assisted multi-camera collaboration system, which enables a camera to efficiently cooperate with the others by sharing the pose. The authors in [33] presented RSPT, a novel active object tracking framework that innovates by reconstructing environmental structure and predicting target trajectories, addressing the challenge of generalization in complex and dynamic scenarios. In [11], the authors introduced the KURL model, which leverages knowledge-guided reinforcement learning, incorporating state recognition and world models to enhance tracking accuracy and efficiency in high-altitude environments. These previous works assume that the cameras can share the information in real time. In fact, the camera's communication delay is not negligible and impacts the collaboration performance of multi-camera AOT, especially for the high-speed moving target. In this work, we consider the effect of communication delay and optimize the communication resource allocation to minimize this communication delay.

2.2 Multi-agent Communication

In the multi-agent system, the communication allows agents to share their local observation information and collaboratively complete a task. As the number of agents increases, the efficient communication becomes more essential.

In the earlier works [25, 16, 21, 10], the communication strategies are manually specified, which are less flexible and difficult to deal with the dynamic tasks. In CommNet [24], a neural network was designed to allow agents to learn the communication relationship among them alongside their policy for decision-making tasks. BiCNet in [19] connected all agents with a Bi-directional LSTM to integrate agent-specific information. ATOC in [7] adopted an attention mechanism to learn when each agent communicates and how to integrate the shared information. Furthermore, the multi-agent learning communication is designed for the perception task. When2com in [12] exploited the attention mechanism to the construct communication group via a handshake process. In particular, each agent first determines whether to require information from others. If it requires, the agent sends a query to determine whom to connect with. Based on that, where2comm [5] introduced the spatial confidence map, which allows agents to share more compact perception information. This approach reduces the communication overhead while maintaining the perceptual accuracy. How2comm [28] employed a flow-guided delay compensation strategy to predict and align future features from collaborators, addressing the challenge of temporal asynchrony caused by transmission delays. Effi-MAOT [29] proposed a learnable communication strategy with a switching mechanism for communication efficient collaborative active object tracking. These works only consider the homogeneous and static communication ability between a pair of agents. In practice, the variable channel results in the heterogeneous and dynamic communication ability. In this paper, we propose a channel-aware collaborative AOT system to dynamically adjust the communication resource allocation.

3 Preliminaries

We present some preliminary knowledge in this section, which includes the system model and the wireless channel model.

3.1 System Model

We consider the system, where N rotatable cameras collaborate to track a target. The cameras are located in given positions and rotate their angles with the moving target. To make each camera continuously track the target, the camera has three modules: an observation sequence encoding module, a communication decision module, and an action decision module. At a time step, each camera first independently captures an image as its local observation. The observation sequence encoding module extracts the target's feature information from the current and historical observations. Then, based on the local feature information, each camera decides whether it can track the target accurately. If not, the camera, called the degraded camera, broadcasts its query and obtains the matching scores with other cameras by employing the attention mechanism. Based on the matching scores and wireless channels, the degraded camera determines its communication graph and resource allocation to efficiently obtain features from other cameras. This determination process is performed in the communication decision module. Here, the communication graph indicates the set of supporting cameras that send features to the degraded camera, while the communication resource allocation includes bandwidth allocation and transmitted message compression ratio for each supporting camera. With the determined communication graph and resource allocation, the cameras share their features via wireless communication. By fusing local observations and obtained features from other cameras, each camera decides its rotating angle in the action decision module.

3.2 Communication Channel

The cameras are connected to share features via a wireless communication network. In a wireless network, the camera's transmission rate depends on many factors, such as the allocated bandwidth, transmit power, and channel. According to the Shannon capacity of wireless channel[26], the transmission rate from camera j to camera i is given by

$$R_{i,j} = B_{i,j}\log_2\left(1 + \frac{P_{i,j} * |h_{i,j}|^2}{N_0}\right), \tag{1}$$

where $B_{i,j}$ is the allocate bandwidth (Hz), $P_{i,j}$ is the transmit power (W) from camera j to camera i, N_0 is the additive white Gaussian noise. Here, $h_{i,j}$ is the channel fading coefficient from camera j to camera i. We model the channel fading coefficient as $h_{i,j} = \sqrt{\beta_{i,j}}g_{i,j}$, where $\beta_{i,j}$ is the large-scale fading component and $g_{i,j} \sim \mathcal{CN}(0,1)$ is the Rayleigh block fading component. The large-scale fading component $\beta_{i,j}$ is relative to the distance between camera j and camera i. The communication delay is

$$d_{j\to i} = \frac{s_{j\to i}\epsilon^t_{j\to i}}{R_{i,j}}, \tag{2}$$

where $s_{j\to i}$ is the size of original transmitted message in bits and $\epsilon^t_{j\to i}$ is the compression ratio from camera j to camera i.

Due to the variability in channel fading coefficients, the cameras' transmission rates are dynamic and heterogeneous. Thus, it is crucial to carefully construct the communication graph, design bandwidth allocation, and the feature compression ratio, thereby improving the collaborative tracking performance. The qualitative relationship among communication graph, resource allocation and collaborative tracking performance is illustrated in Fig. 2. Both allocated bandwidth and feature compression ratio affects the communication delay for a pair of cameras. Combined with the communication graph, this determines the communication delays for all connected cameras, thereby leading to different tracking performance.

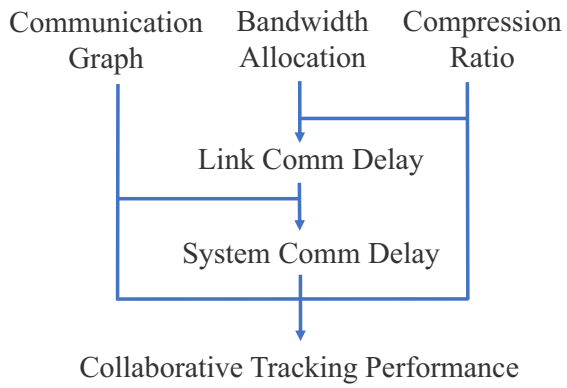


Figure 2: The qualitative relationship among communication graph, resource allocation and collaborative tracking performance.

4 Problem Formulation

In the considered system model, N rotatable cameras collaboratively track the target during T time steps. Denote C^t_i and O^t as the observation of camera i and the target position at time step t, respectively. The objective of multi-camera active object tracking is to maximize the average tracking success rate over T time steps under the total bandwidth budget B, given by

$$\begin{aligned} \arg\max_{\theta,\mathbf{p}} \quad & \frac{1}{T}\sum_{t=1}^{T} \mathrm{ASR}(\mathbf{C}^t, O^t) \\ \text{s.t.} \quad & \sum_{j=1}^{N} b^t_{j\to i} \leqslant B, \quad \forall i \in \{1,2,3,\ldots,N\} \end{aligned} \tag{3}$$

where $\mathbf{C}^t = \{C^t_1, ..., C^t_N\}$ and $\mathrm{ASR}(\mathbf{C}^t, O^t)$ is expressed as

$$\begin{aligned} &\mathrm{ASR}(\mathbf{C}^t, O^t) \\ &= \frac{1}{N}\sum_{i=1}^{N} g\left(f_{\theta,\mathbf{p}}\left(C^t_i, \{(m^t_{j\to i}, d^t_{j\to i})\}_{j=1}^{N}\right), O^t\right), \end{aligned} \tag{4}$$

where $m^t_{j\to i}$ is the camera i's received packet from camera j, $d^t_{j\to i}$ is the transmission delay from camera j to camera i, $f(\cdot,\cdot)$ is the rotation action decision-making network and $g(\cdot,\cdot)$ is the tracking accuracy. Here, the optimized parameters include the trainable parameter θ of rotation action decision-making network and the resource allocation parameter set $\mathbf{p}^t_{j\to i} = \{\epsilon^t_{j\to i}, b^t_{j\to i}\}, \forall i, j \in \{1, ..., N\}$ and $\forall t \in \{1, ..., T\}$, where $\epsilon^t_{j\to i}$ and $b^t_{j\to i}$ represent the compression ratio and the allocated bandwidth of transmitted packet from camera j to camera i at time step t, respectively. By optimizing the network parameter and resource allocation, the average tracking success rate is maximized for a given bandwidth budget.

Note that, when obtaining the optimized parameters $\epsilon^t_{j\to i} = b^t_{j\to i} = 0$, it indicates no collaboration from camera j to camera i. Thus, optimizing the parameters $\epsilon^t_{j\to i}$ and $b^t_{j\to i}$ implies the optimization of both collaboration relationship and allocated transmission resources. Moreover, apart from the observation complementary employed in the most studies on multi-camera AOT, the system design in this paper considers the effect of wireless channel fading on the communication delay, thereby improving the collaborative tracking performance. In particular, the resource allocation, including the bandwidth and compression rate, affects the communication delay between two cameras, while the collaboration relationship affects the delay distribution in the network. As the communication delay varies with the channel, as given by (1) and (2), this system design can adapt to varying channels by dynamically adjusting allocated resources and collaboration relationship.

The problem in (3) is transformed to a Decentralized Partially Observable Markov Decision Process (Dec-POMDP)[2, 18]. Define a tuple $M_{DecP} = \langle N, S, A, P, R, O, Z, \gamma\rangle$, where N is the number of agents, S is the state space, $A = \times_{i\in N} A_i$ is the set of joint actions, P is the transition probability function, R is the reward function, $O = \times_{i\in N} O_i$ is the set of joint observations, Z is the observation probability function, and γ is the discount factor. At time step t, the set of joint actions is denoted by $a_t =< a_{1,t}, a_{2,t}...a_{N,t} >$, where $a_{i,t}$ is the action of agent i. Similarly, $o_t =< o_{1,t}, o_{2,t}...o_{N,t} >$ is the set of joint observations at time step t, where $o_{i,t}$ is the observation of agent i.

Based on the observation $o_{i,t}$, agent i decides its strategy based on the policy $\pi_i(a_{i,t}|o_{i,t})$, which implies a probability distribution over the action $a_{i,t}$. When agents execute their actions based on their policies, the state becomes s_{t+1} and the observation is obtained as s_{t+1} according to the state transition function $P(s_{t+1} \mid s_t, a_t)$ and the observation likelihood function $O(o_{t+1} \mid s_{t+1}, a_t)$, respectively. The global reward at time step t is given by $R_t = R(s_t, a_t)$. For a cooperative task, the goal is to maximize the cumulative global reward by optimizing each agent's policy, that is

$$E_{\pi_1,\ldots,\pi_N}\left[\sum_{t=1}^{T} R_t\right]. \tag{5}$$

5 CAMAOT: Channel-Aware Multi-Camera Active Object Tracking System

CAMAOT is designed to solve the formulated optimization problem. As shown in Fig. 3, the network of each camera is comprised of an observation sequence encoding module, a communication decision module, and an action decision module. The camera extracts features from its local observation in the observation sequence encoding module. Based on local features and received information from other cameras, the camera determines its communication graph and resource allocation in the communication decision module. With this communication graph and resource allocation, the agent receives features from its collaborators and decides the rotation action via the action decision module.

5.1 Observation Sequence Encoding Module

In this module, we develop an observation encoder f_o and a sequence encoder f_s. The observation encoder processes the local observation

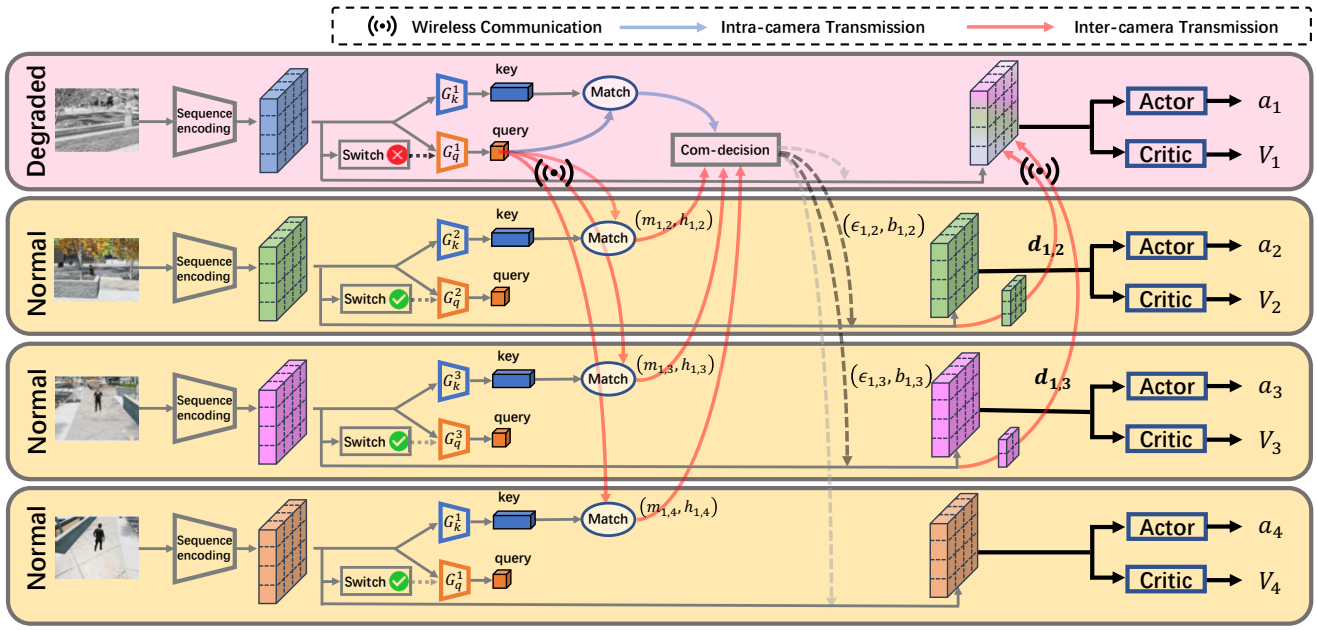


Figure 3: The architecture of our channel-aware multi-camera AOT system. Cameras initially determine the need for assistance based on their local observations. Subsequently, degraded cameras broadcast queries and, upon receiving matching scores and channel conditions, their communication decision modules construct communication graphs and determine communication resource allocation. Based on these decisions, they receive and integrate features from selected cameras. Finally, all cameras select rotational actions based on the integrated features to track the target.

o_i^t to extract features ϕ_i^t at the current time step. Subsequently, the observation history $H_i^t = \{\phi_i^1, \phi_i^2, ...\phi_i^t\}$ is input into the sequence encoder to obtain the timing feature ψ_i^t, characterising the target's temporal dynamics, such as velocity and trajectory. We employ Convolutional Neural Network (CNN) and LSTM networks to construct the encoder $f_o(\cdot)$ and $f_s(\cdot)$, respectively.

5.2 Communication Decision Module

To achieve the wireless channel-aware collaboration among cameras, the communication decision module determines the communication graph and allocated resources of corresponding collaborators. First, we design a switch based on a binary classification neural network, whose input is the timing feature $\psi_{n,t}$, to determine whether the camera requires information from other cameras.

When it requires information from other cameras, the camera decides the collaborators and resource allocation via a handshake process. This process consists of querying, matching, and decision-making stages. In the querying stage, camera i generates a Q-dimensional query at time step t, given by

$$\mu_i^t = G_q\left(\psi_i^t, \theta_q\right) \in \mathbb{R}^Q, \tag{6}$$

and a K-dimensional key κ_i^t, given by

$$\kappa_i^t = G_k\left(\psi_i^t, \theta_k\right) \in \mathbb{R}^K, \tag{7}$$

where $G_q(\cdot)$ is the query generator and $G_k(\cdot)$ is the key generator. Based on that, the camera broadcasts its query to other cameras.

In the matching stage, by exploiting the scaled general attention[3, 15, 27], the camera obtains a matching score between the received query and its own key, calculated as

$$m_{i,j} = \Phi\left(\mu_i, \kappa_j\right) = \frac{\mu_i^t W_g \kappa_j}{\sqrt{K}}, \tag{8}$$

where $W_g \in \mathbb{R}^{Q\times K}$ is a learnable parameter. Then, the camera sends its matching score to the querying camera. It is noteworthy that when camera i does not broadcast its query, we have $m_{i,i} = 1$ and $m_{i,j} = 0$ for $\forall k \neq n$. Once all the matching scores have been calculated, the matrix M is formed, whose element in row i and column j is $m_{i,j}$.

After receiving the matching scores and channel fading coefficients from all other cameras, the camera determines its collaborators and their allocated resources by solving the optimization problem in (3). Due to the optimization over a time sequence, it is non-trivial to solve this problem via conventional optimization methods. Thus, we employ multi-agent reinforcement learning to solve this optimization problem in this paper, where each camera corresponds to an agent and learns the optimal resource allocation strategy by interacting with the environment.

5.3 Action Decision Module

With the determined communication graph and resource allocation, the cameras share their features via wireless communication. After receiving features from the collaborators, camera i concatenates these received features and its own features, given by

$$f_{i,t} = \left[\psi_{i,t}; \psi_{i,t}^{int}\right] \tag{9}$$

where $[\cdot;\cdot]$ is the concatenation operator and $\psi_{i,t}^{int}$ is obtained by

$$\psi_{i,t}^{int} = \sum_{j=1,\neq i}^{N} \bar{m}_{i,j} \psi'_{j,t}. \tag{10}$$

where weight $\bar{m}_{i,j}$ is the element of the ith row and jth column in a sparse representation of matrix M. The sparse representation $\bar{M}$ is obtained by using the Softmax function in matrix M. $\psi'_{j,t}$ represents the compressed feature of $\psi_{j,t}$ according to ratio $\epsilon^t_{j\to i}$.

The concatenated feature $f_{i,t}$ of camera i is fed into the actor network and the critic network in the action decision module. The actor network outputs a policy distribution $\pi_n(a_{n,t}|f_{n,t})$, which is used for sampling tracking decisions. The critic network outputs a value function $V(f_{n,t})$ for the corresponding action. Based on two outputs, the network parameters is updated until reaching the convergence.

5.4 Training CAMAOT

Reward function. For the multi-camera cooperative tracking task, the aim is to keep the target within the field of view of each camera. Thus, the tracking reward function of camera i at time step t is defined as:

$$R^a_{i,t} = \begin{cases} 2 - \frac{\Delta\alpha_t}{\alpha_{max}} - \frac{\Delta\beta_t}{\beta_{max}} & (a) \\ 0 & (b) \\ -1 & (c) \end{cases} \tag{11}$$

where $\Delta\alpha_t$ and $\Delta\beta_t$ are the absolute pitch angle error and the absolute yaw angle error between the camera's direction and the target's direction, respectively. Here, α_{max} and β_{max} are the maximum control bound of corresponding angle errors. In Equation (11), (a) means that the target is visible in the image, (b) means that obstacles occlude the target, and (c) means that the target is outside the view.

Furthermore, to mitigate the effect of intolerantly long communication delay on the tracking performance, a maximum communication delay threshold D_{max} is introduced. In this case, the camera can only receive features from the cameras whose transmission time is smaller than this threshold. To receive more features within the duration D_{max}, a penalty is defined to the communications whose delays are larger than D_{max}, given by

$$R^b_{i,t} = \begin{cases} 1 - \frac{q_{i,t}}{Q_{i,t}} & (a) \\ 0 & (b) \end{cases} \tag{12}$$

where $Q_{i,t}$ is the number of camera i's collaborators at time step t. Among these collaborators, $q_{i,t}$ collaborators' communication delays are larger than D_{max}. Similarly to (11), (a) in (12) indicates that the target is visible in the image and (b) indicates that the target is outside the image. Therefore, the cumulative global reward at time step t is obtained as

$$R_t = \sum_{i=1}^{N} (R^a_{i,t} + R^b_{i,t}). \tag{13}$$

Training strategy. To optimize each agent's policy to maximize the cumulative global reward, we modify the conventional RL algorithm, i.e., A3C[17], by adding the communication module. For the modified RL algorithm, we adopt a centralized training and decentralized inference paradigm[13, 6]. In particular, during the training phase, we consider that each camera can be connected to all other cameras. The global observation is obtained by concatenating received features with weights, where the weights are obtained by using (8). The concatenated features are fed into the actor network and the critic network, to obtain the action distribution and the corresponding values, respectively. These values are used to update the network parameters. Note that, the switch in the communication module is trained by using the binary cross entropy loss. A well-trained switch accurately implies the camera observation status. In other words, when the camera's observation is well enough, its switch is closed. Otherwise, its switch is open for receiving other cameras' features.

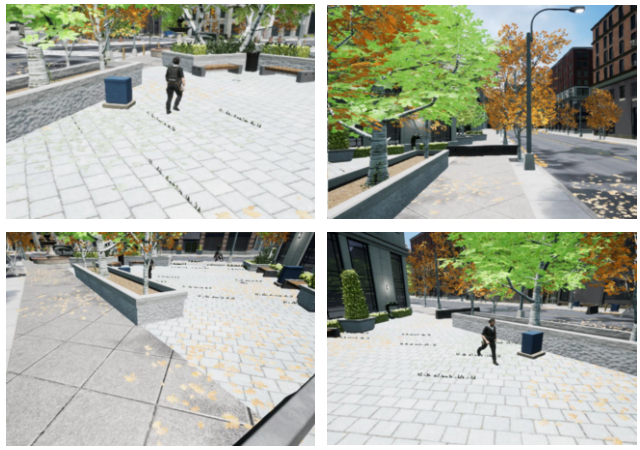


Figure 4: Example scene in Urban City.

6 Experimental Results

In this section, we conduct the experiments in a high-fidelity simulation based on Unreal Engine 4. By comparing to the considered baselines, we demonstrate the performance gain from CAMAOT.

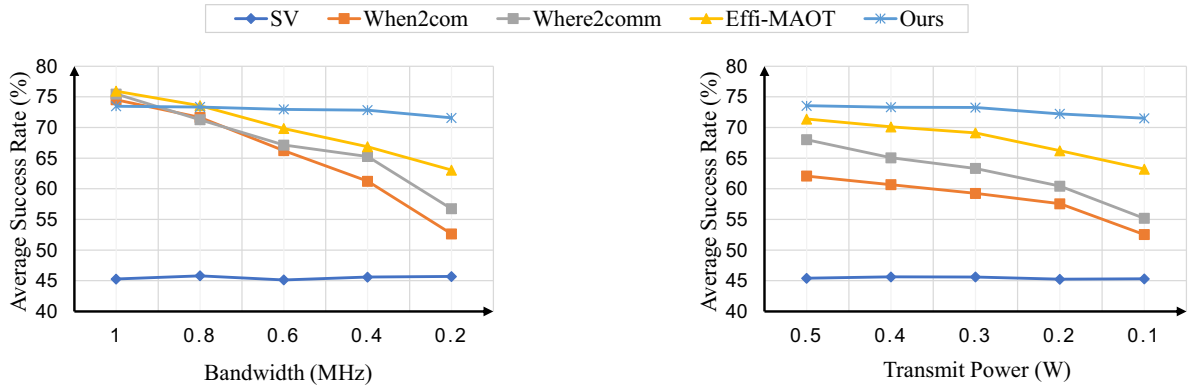
6.1 Experimental Settings

We adopt Unreal Engine 4[23, 8] to construct a high-fidelity simulation environment of *Urban City* and use gym-unrealcv[20, 31] to provide a convenient interface with reinforcement learning, as shown in Fig. 4. In this environment setting, the target moves around a flower bed at a given speed and four rotatable cameras are placed at fixed positions to perform active tracking. Each camera captures an RGB image as its observation and extracts features from this image. By sharing features and pose information via wireless communication, the cameras decide their actions. The objective of this task is to continuously track the target for each camera. The camera's discrete action space includes nine components, i.e., turn left, turn right, turn up, turn down, turn top-left, turn top-right, turn bottom-left, turn bottom-right, and keep still. The wireless channel between any pair of cameras follows a Rayleigh fading, as described in Section 3.2. According to the 3GPP standard [1], the large-scale fading component of wireless channel between camera i and camera j is given by

$$\beta_{i,j} = 10 * 3.76 * \log_{10}\frac{l_{i,j}}{1000} + 128.1, \tag{14}$$

where $l_{i,j}$ is the distance between camera i and camera j. The noise power density over the channel is -196 dBm/Hz. Each camera's transmit power is 20dBm.

Note that, in order to learn a good feature representation of each camera's observation at the training process, the randomization method is employed as environment augmentation[22]. In particular, both the target's initial position and movement speed are random. We also note that the well-trained CAMAOT system can be applied in the real-world environment, which needs to be further investigated as the future work.



(a) Comparison results at different channel bandwidths

(b) Comparison results at different transmission powers

Figure 5: Comparison results under different communication resources.

6.2 *Evaluation Metrics and Baselines*

We define *Success Rate* to evaluate the tracking performance of each camera, given by

$$P_{s,i} = \frac{1}{T}\sum_{t=1}^{T} D_{i,t}, \tag{15}$$

where $D_{i,t}$ indicates whether the target is located within camera i's view at time step t. When it is within the view, $D_{i,t}$ equals one. Otherwise, $D_{i,t}$ equals zero. It is shown that $P_{s,i}$ represents the probability that the target is located within camera i's view at a time step. Based on that, we define the success rate averaged over all cameras, named as *Average Success Rate* (ASR), to evaluate the system performance, i.e.,

$$P_s = \frac{1}{N}\sum_{i=1}^{N} P_{s,i}. \tag{16}$$

We consider an independent perception system and several collaborative perception systems with distributed communication as our baselines.

- *Independent Perception* (**SV**): Each camera only relies on its own observation to determine the rotation angle. There is no communication among cameras.
- *When2com*: Each camera employs the attention mechanism to select a set of other cameras as collaborators. The camera fuses its observation and those received from collaborators for perception.
- *Where2comm*: The camera has a spatial confidence map, which reflects the spatial heterogeneity of its observation. Based on that, the camera obtains spatially sparse but critical information for sharing, thereby improving the communication efficiency.
- *Effi-MAOT*: The camera needs to determine whether to receive information from other agents or use its own observation alone, before selecting collaborators. Moreover, the camera extracts features from both the current observation and historical observations.

6.3 *Quantitative Evaluation*

Benchmark comparison. Table 1 compares the success rate of the proposed CAMAOT and that of baselines, when the target's movement speed is 3m/s and each camera's available bandwidth is 0.1MHz. It can be seen that the proposed CAMAOT achieves a significant improvement over all the considered baselines, in terms of the single camera's success rate and ASR. In particular, compared to Where2comm and Effi-MAOT, CAMAOT improves ASR by 16.33% and 7.2%, respectively. This performance gain is achieved by the careful design of resource allocation for dynamic wireless channels.

Table 1: Success rate comparison for the given movement speed and bandwidth.

Cam_id	Success Rate(%)				
	SV	When2com	Where2comm	Effi-MAOT	Ours
Cam_1	44.12	54.01	55.82	55.65	**61.33**
Cam_2	63.93	64.14	64.92	65.73	**72.61**
Cam_3	32.89	46.18	49.96	68.16	**77.56**
Cam_4	40.30	45.83	50.01	67.70	**74.54**
ASR	45.31	52.54	55.18	64.31	**71.51**

Robustness to target's movement speed. We compare ASR of the proposed CAMAOT with that of baselines for various target's movement speeds, as shown in Table 2. We see that the proposed CAMAOT enhances ASR significantly across all the target's speed choices. When the target's speed is 4m/s, CAMAOT achieves the ASR enhancement of 9.38%, compared to Effi-MAOT. Moreover, it can be seen from the table that CAMAOT is more robust to the change of target's movement speed, while the tracking performance of all the models degrades with an increase of the target's movement speed. For example, when the target's movement speed increases from 1m/s to 4m/s, ASR of the proposed CAMAOT decreases by 5.66%, while ASR of Where2comm and Effi-MAOT reduces by 17.12%, and 10.55% respectively. This is because that the optimized resource allocation in CAMAOT, which is adaptive to the wireless channel, decreases the overall transmission time, thereby mitigating the impact of cameras' delayed rotation, especially for the large movement speed.

Robustness to communication resources. Fig. 5 shows the effect of different available communication resources, i.e., bandwidth and transmit power, on the tracking performance. For SV, there is no communication among cameras. Then, the tracking performance of SV is constant for the varying amount of communication resources.

The effect of available bandwidth B on the tracking performance is shown in Fig. 5(a). We see that compared to the baseline systems,

Table 2: Comparison results of average success rate at different target movement speeds.

Models	1m/s	2m/s	3m/s	4m/s
SV	58.31	49.70	45.31	43.74
When2com	69.47	62.26	52.54	50.91
Where2comm	68.58	62.17	55.18	51.46
Effi-MAOT	71.84	65.08	64.31	61.29
Ours	**76.33**	**72.45**	**71.51**	**70.67**

the proposed CAMAOT achieves a superior ASR over all the available bandwidth choices. Moreover, this superiority is more obvious for the low to medium amount of available bandwidth. In particular, for $B = 0.6$MHz and $B = 0.2$MHz, the proposed CAMAOT improves ASR by 3.13% and 8.52%, respectively, compared to Effi-MAOT. This is because that a small amount of bandwidth results in the long transmission time, thereby seriously deteriorating the tracking performance. In this case, the optimized bandwidth allocation in CAMAOT can reduce the transmission time of some critical information and mitigate the impact of small amount of bandwidth. Furthermore, it is shown from Fig. 5(a) that in order to obtain a given ASR, the proposed CAMAOT requires the least bandwidth. When obtaining the ASR of 70%, the required bandwidth of CAMAOT is reduced by 66.7% and 75%, compared to Effi-MAOT and Where2com, respectively.

Fig. 5(b) shows the effect of transmit power on the tracking performance. It can be seen that the proposed CAMAOT achieves a performance improvement for all the transmit powers. Similarly to the effect of available bandwidth, this improvement is more significant for the low to medium transmit power. Moreover, as the amount of transmit power reduces, the decrease of ASR in the proposed CAMAOT is much slower than that of baseline systems. It implies the better robustness to the reduced transmit power. For example, When the transmit power reduces from 0.5W to 0.1W, ASR of When2comm, Where2comm, and Effi-MAOT decrease by 9.54%, 12.84%, and 9.07% respectively. The ASR reduction of the proposed system is only 2.04%. In addition, the proposed CAMAOT can achieve the same tracking performance with much smaller transmit power than the baseline systems, which is very beneficial for the cameras with limited battery capacity. When the tracking performance target is 71%, CAMAOT reduces the transmit power by 5 times, compared to Effi-MAOT.

6.4 Ablation Studies

To investigate the necessity of different designs in CAMAOT, we perform thorough ablation studies. In particular, three following questions require to be answered: 1) Is dynamic bandwidth allocation necessary? 2) Is it crucial to dynamically modify the information compression rate? 3) Is concurrent adjustment of both bandwidth allocation and information compression rates necessary? To this end, we present three resouce allocation systems:

- **FixedE&EqualB**: Cameras select their collaborators as the process in CAMAOT. Then, they share information with collaborators under a given compression ratio and an equally allocated bandwidth.
- **FixedE**: Cameras share information with their selected collaborators under a given compression ratio and an optimally allocated bandwidth.
- **EqualB**: Cameras share information with their selected collaborators under an optimal compression ratio and an equally allocated bandwidth.

Table 3: Ablation study results of the proposed CAMAOT.

Cam_id	Success Rate(%)			
	FixedE&EqualB	FixedE	EqualB	CAMAOT
Cam_1	45.65	54.91	53.48	**61.33**
Cam_2	53.82	58.49	58.25	**72.61**
Cam_3	57.41	62.77	62.87	**77.56**
Cam_4	60.36	64.67	62.47	**74.54**
ASR	54.31	60.21	59.27	**71.51**

Effect of bandwidth allocation optimization. By comparing the tracking performance of FixedE and FixedE&EqualB in Table3, we can see that FixedE achieves a higher ASR. This is because that the optimal bandwidth allocation can mitigate the effect of varying wireless channels on the transmission time, especially for the critical information, so as to improve the tracking performance.

Effect of compression ratio optimization. In Table3, we see that EqualB improves the tracking performance, compared to FixedE&EqualB. This verifies the necessity of compression ratio optimization. The compression ratio optimization can be adaptive to the importance of information, to reduce the bandwidth requirement of less important information transmissions.

Effect of joint bandwidth allocation and compression ratio optimization. In Table3, we see that the proposed CAMAOT achieves the highest ASR, compared to other three systems. In particular, the proposed CAMAOT achieves an ASR improvement of 11.3% and 12.24% over FixedE and EqualB, respectively. This gain is from the joint bandwidth allocation and compression ratio optimization in CAMAOT. By taking the effects of wireless channels and information importance into account, this joint optimization can be adaptive to the environment dynamics.

The results presented in Table 3 demonstrate the performance advantage of the proposed CAMAOT. In particular, the design of communication graph, information compression, and bandwidth allocation strategies in the communication decision module can ensure that the critical information is transmitted with a low delay for varying wireless channels, to improve the tracking performance.

7 Conclusion

This paper proposed a novel multi-camera active object tracking system, to overcome the challenges associated with time-varying communication environments encountered in the practical intelligent surveillance. The proposed channel-aware multi-camera AOT (CAMAOT) system enhanced collaborative tracking by dynamically adjusting the communication graph and resource allocation in response to changing channel conditions. The RL based implementation of this communication decision module enabled the system to efficiently manage communication among cameras, ensuring that essential tracking information is promptly shared even under bandwidth limitations. Extensive experiments in environments that simulate real-world conditions, have confirmed that CAMAOT consistently outperforms existing systems, in terms of the tracking performance and the communication resource consumption. These results verified the effectiveness of our proposed system in addressing the complexities of real-time, multi-camera tracking in intelligent surveillance systems.

Acknowledgements

We would like to thank the reviewers for their comments, which helped improve this paper considerably. This work was partially supported by the National Science Fund for Distinguished Young Scholars (62025205) and the National Natural Science Foundation of China (No. 62032020, No. 62102322).

References

[1] D. 3GPP. Study on new radio access technology physical layer aspects. *Technical Report (TR) 38.802, V14. 2.0*, 2017.

[2] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4):819–840, 2002.

[3] J. Cheng, L. Dong, and M. Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.

[4] Z. Fang, J. Zhao, M. Yang, W. Zhou, Z. Lu, and H. Li. Coordinate-aligned multi-camera collaboration for active multi-object tracking. *arXiv preprint arXiv:2202.10881*, 2022.

[5] Y. Hu, S. Fang, Z. Lei, Y. Zhong, and S. Chen. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. *Advances in neural information processing systems*, 35: 4874–4886, 2022.

[6] T. Ikeda and T. Shibuya. Centralized training with decentralized execution reinforcement learning for cooperative multi-agent systems with communication delay. In *2022 61st Annual Conference of the Society of Instrument and Control Engineers (SICE)*, pages 135–140. IEEE, 2022.

[7] J. Jiang and Z. Lu. Learning attentional communication for multi-agent cooperation. *Advances in neural information processing systems*, 31, 2018.

[8] H. Lee, S. Ryoo, and S. Seo. A comparative study on the structure and implementation of unity and unreal engine 4. *Journal of the Korea Computer Graphics Society*, 25(4):17–24, 2019.

[9] J. Li, J. Xu, F. Zhong, X. Kong, Y. Qiao, and Y. Wang. Pose-assisted multi-camera collaboration for active object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 759–766, 2020.

[10] Y. Li, B. Bhanu, and W. Lin. Auction protocol for camera active control. In *2010 IEEE International Conference on Image Processing*, pages 4325–4328. IEEE, 2010.

[11] X. Liu, J. Tan, X. Ren, W. Ren, and H. Dai. Kurl: A knowledge-guided reinforcement learning model for active object tracking. In *Asian Conference on Machine Learning*, pages 818–833. PMLR, 2024.

[12] Y.-C. Liu, J. Tian, N. Glaser, and Z. Kira. When2com: Multi-agent perception via communication graph grouping. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 4106–4115, 2020.

[13] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

[14] W. Luo, P. Sun, F. Zhong, W. Liu, T. Zhang, and Y. Wang. End-to-end active object tracking and its real-world deployment via reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 42(6):1317–1332, 2019.

[15] M.-T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[16] F. S. Melo, M. T. Spaan, and S. J. Witwicki. Querypomdp: Pomdp-based communication in multiagent systems. In *Multi-Agent Systems: 9th European Workshop, EUMAS 2011, Maastricht, The Netherlands, November 14-15, 2011. Revised Selected Papers 9*, pages 189–204. Springer, 2012.

[17] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[18] F. A. Oliehoek, C. Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

[19] P. Peng, Y. Wen, Y. Yang, Q. Yuan, Z. Tang, H. Long, and J. Wang. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2017.

[20] W. Qiu and A. Yuille. Unrealcv: Connecting computer vision to unreal engine. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 909–916. Springer, 2016.

[21] F. Qureshi and D. Terzopoulos. Smart camera networks in virtual reality. *Proceedings of the IEEE*, 96(10):1640–1656, 2008.

[22] J. Salamon and J. P. Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal processing letters*, 24(3):279–283, 2017.

[23] A. Sanders. *An introduction to Unreal engine 4*. CRC Press, 2016.

[24] S. Sukhbaatar, R. Fergus, et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29, 2016.

[25] M. Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pages 330–337, 1993.

[26] D. Tse and P. Viswanath. *Fundamentals of wireless communication*. Cambridge university press, 2005.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[28] D. Yang, K. Yang, Y. Wang, J. Liu, Z. Xu, R. Yin, P. Zhai, and L. Zhang. How2comm: Communication-efficient and collaboration-pragmatic multi-agent perception. *Advances in Neural Information Processing Systems*, 36, 2024.

[29] M. Yin, Z. Sun, B. Guo, and Z. Yu. Effi-maot: A communication-efficient multi-camera active object tracking. In *2023 19th International Conference on Mobility, Sensing and Networking (MSN)*, pages 9–16. IEEE, 2023.

[30] K. Zhang, Z. Yang, and T. Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pages 321–384, 2021.

[31] F. Zhong, W. Qiu, T. Yan, Y. Alan, and Y. Wang. Gym-unrealcv: Realistic virtual worlds for visual reinforcement learning. Web Page, 2017. URL https://github.com/unrealcv/gym-unrealcv.

[32] F. Zhong, P. Sun, W. Luo, T. Yan, and Y. Wang. Ad-vat+: An asymmetric dueling mechanism for learning and understanding visual active tracking. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1467–1482, 2019.

[33] F. Zhong, X. Bi, Y. Zhang, W. Zhang, and Y. Wang. Rspt: reconstruct surroundings and predict trajectory for generalizable active object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 3705–3714, 2023.